

KIET JOURNAL OF COMPUTING AND INFORMATION SCIENCES



ISSN (P): 2616-9592
ISSN (E): 2710-5075



Volume:7

Issue: 1

Jan - June

2024



KIET
JOURNAL
OF COMPUTING AND
INFORMATION SCIENCES

Volume 7, Issue 1, 2024

ISSN (P): 2616-9592

ISSN (E): 2710-5075

Frequency Bi-Annual

Editorial Board

Patron

Air Vice Marshal (Retd) Usaid Ur Rehman Usmani, SI(M), TI(M) - President, KIET

Editor-in-Chief

Prof. Dr. Muzaffar Mahmood

Associate Editor

Prof. Dr. Maaz bin Ahmed

Managing Editor

Prof. Dr. Muhammad Khalid Khan

Manager Production & Circulation

Mr. Saad Khan

College of Computing & Information Sciences
Karachi Institute of Economics & Technology



College of Computing & Information Sciences

Vision

To develop technology entrepreneurs & leaders for national & international market

Mission

To produce quality professionals by using diverse learning methodologies, aspiring faculty, innovative curriculum and cutting edge research, in the field of computing & information sciences.





AIMS AND SCOPE

KIET Journal of Computing and Information Sciences (KJCIS) is the bi-annual, multi-disciplinary research journal published by **College of Computing & Information Sciences (CoCIS)** at **Karachi Institute of Economics and Technology (KIET)**, Karachi, Pakistan. **KJCIS** aims to provide a panoramic view of the state of the art development in the field of computing and information sciences at global level.

It provides a premier interdisciplinary platform to researchers, scientists and practitioners from the field of computing and information sciences to share their findings and contribute to the knowledge domain at global level. The journal also fills the gap between academician and industrial research community.

KJCIS focused areas for publication includes; but not limited to:

- Data mining
- Big data
- Machine learning
- Artificial intelligence
- Mobile applications
- Computer networks
- Cryptography and information security
- Mobile and wireless communication
- Adhoc and body area networks
- Software engineering
- Speech and pattern recognition
- Evolutionary computation
- Semantic web and its application
- Data base technologies and its applications
- Internet of things (IoT)
- Computer vision
- Distributed computing
- Grid and cloud computing



OPEN ACCESS POLICY

For the benefit of authors and research community, this journal adopts open access policy, which means that the authors can self-archive their published articles on their own website or their institutional repositories. The readers can download or reuse any article free of charge for research, further study or any other non profitable academic activity.

PEER REVIEW POLICY

Peer review is the process to uphold the quality and validity of the published articles. KJCIS uses double-blind peer review policy to ensure only high-quality publications are selected for the journal. Papers are referred to at least two experts as suggested by the editorial board. All publication decisions are made by the journal's Editors-in-Chief on the basis of the referees' reports. We expect our Board of Reviewing Editors and reviewers to treat manuscripts as confidential material. The identities of authors and reviewers remain confidential throughout the process.

COPYRIGHT

All rights reserved. No part of this publication may be produced, translated or stored in a retrieval system or transmitted in any form or by any means; electronic, mechanical, photocopying and/ or otherwise the prior permission of publication authorities.

DISCLAIMER

The opinions expressed in **KIET Journal of Computing and Information Sciences (KJCIS)** are those of the authors and contributors, and do not necessarily reflect those of the journal management, advisory board and the editorial board. Papers published in KJCIS are processed through double blind peer-review by subject specialists and language experts. Neither the **CoCIS** nor the editors of **KJCIS** can be held responsible for errors or any consequences arising from the use of information contained in this journal, instead; errors should be reported directly to the corresponding authors of the articles.



Academic Editorial Board

Dr. Ronald Jabangwe University of Southern Denmark, Denmark	Dr. Sardar Anisul Haque Alcorn State University, USA
Dr. M. Ajmal Khan Ohio Northern University, USA	Dr. Yasser Ismail Southern University Louisiana, USA
Dr. Suliman A. Alsuhibany Qassim University, Saudi Arabia	Dr. Manzoor Ahmed Hashmani University of Technology Petronas, Malaysia
Dr. Wael M El-Medany University of Bahrain, Bahrain	Dr. Atif Tahir FAST NUCES, Pakistan
Dr. Asim Imdad Wagan Mohammad Ali Jinnah University, Pakistan	Dr. Affan Alim Iqra University, Karachi
Dr. Salman A. Khan Karachi Institute of Economics & Tech, Pakistan	Dr. Taha Jilani Baharia University, Karachi

Advisory Board

Dr. Andries Engel brecht University of Pretoria, South Africa	Dr. Mohamed Amin Embi University Kebangsaan, Malaysia
Dr. Rashid Mehmood King Abdul Aziz University, Saudi Arabia	Dr. Anh Nguyen-Duc Norwegian University of Technology, Norway
Dr. Ibrahima Faye University of Technology Petronas, Malaysia	Dr. Tahir Riaz Data Architect, SleeknoteApS, Denmark
Dr. Faraz Rasheed Microsoft, USA	Dr. Mostafa Abd-El-Barr Kuwait University, Kuwait
Dr. Abdul Naser Mohamed Rashid Qassim University, Saudi Arabia	Dr. Mohd Fadzil Bin Hassani University of Technology Petronas, Malaysia
Dr. Syed Irfan Hyder Ziauddin University, Pakistan	Dr. Bawani S. Chowdry Mehran University, Jamshoro, Pakistan
Dr. Jawad Shami FAST - NUCES, Pakistan	Dr. Nasir Tauheed Institute of Business Administration, Pakistan



Table of Content

1 01-13	Power Optimized Task Scheduling using Genetic Algorithm (POTS-GA) in Cloud Environment <i>Sana Saleem, Minhaj Ahmad Khan</i>
	Accurate Attack Detection in Intrusion Detection System for cyber Threat Intelligence Feeds using Machine Learning Techniques <i>Ehtsham Irshad, Abdul Basit Siddiqui</i>
2 14-34	
3 35-50	Predicting Student Performance using Educational Data Mining: A Review <i>Veena Kumari, Areej Fatemah Meghji, Rohma Qadir, Urooj Oad</i>
	A Hybrid Model for Human Behavior Recognition using Emotions, Sentiments, and Mood Features <i>Asia Samreen, Syed Asif Ali, Hina Shakir, Muhammad Hussain</i>
4 51-66	
5 67-84	Predicting and Characterizing piRNAs and their Functions using Integrated Machine Learning Approach <i>Anam Umera, Sajid Mahmood, Usman Inayat</i>

Power Optimized Task Scheduling using Genetic Algorithm (POTS-GA) in Cloud Environment

Sana Saleem^a, Minhaj Ahmad Khan^{a*}

^aBahauddin Zakariya University, Multan, Pakistan
ssaleem3399@gmail.com, mik@bzu.edu.pk

*Corresponding Author: Minhaj Ahmad Khan mik@bzu.edu.pk

Abstract

In a cloud environment, the allocation of tasks has become pivotal on account of rapid growth of user requests. The processing of user requests leads to a significant execution time, and a huge amount of power is also consumed. Consequently, task scheduling for optimizing makespan and power usage has become critical, particularly in a heterogeneous environment. This research work proposes Power-Optimized Task Scheduling using Genetic Algorithm (POTS-GA) that aims to minimize execution time and power consumption. The proposed strategy employs genetic algorithm to take scheduling decision while taking into consideration the execution time and overall power consumption of resources. The fitness computation considering both objectives and the customized genetic operators ensure to search for a better scheduling solution. The experiments performed on a large number of tasks and virtual machines show that the proposed POTS-GA approach outperforms other task scheduling strategies including Efficient Task Allocation using Genetic Algorithm (ETA-GA), Round Robin algorithm (RRA), First Come First Serve (FCFS) and Greedy algorithm in terms of makespan and power consumption.

Keywords: Cloud Computing, Scheduling, GA, Makespan, Power Optimization

1. Introduction

As a cutting-edge technology, cloud computing has revolutionized the digital realm by providing economical solutions for its users. It has changed the computing mechanism by its on-demand delivery of IT resources to its users over internet with pay-per-use model [1]. The ubiquity of cloud computing makes services available over internet from anywhere. Hence, many industries and research organizations have adopted cloud computing technologies for economic benefits [2]. Nowadays, the internet users may access computing services all over the world, without requiring them to think about the hosting infrastructure. Moreover, the capability of internet regarding provision of services through access to resources such as storage, network

and processors, is fully harnessed by cloud computing. This type of hosting architecture is made up of powerful machines that are deployed by service providers for use by their consumers. By offering services to cloud service consumers, the cloud service providers generate revenue, which makes cloud computing a widely used paradigm.

The cloud computing environment generally provides services to its users through a variety of models corresponding to the Infrastructure, Software, and, Platform [3]. Such environment can offer high performance because it distributes workloads over all resources in a fair and efficient manner, resulting in reduced execution times, waiting times, maximum throughput, and efficient resource utilization. Since the demand for cloud services is rapidly growing, the rate of growth for large-scale computing datacenters along with huge amount of high performance resources is also increased. A datacenter in a cloud environment serves as a storage infrastructure for physical resources in order to offer cloud services to clients [4] [5].

The virtual resources that are used to handle the client requests are mapped to physical machines in the cloud computing environment. The virtual machine (VM) placement algorithms in this context are used to identify the appropriate physical machines for mapping to the VMs [6] [7] [8] [9]. The user requests arranged as tasks are allocated to VMs with different computational capacities. An effective task scheduling strategy maps the tasks to appropriate VMs for minimizing overall execution time (or makespan) [10] [11] [12]. During execution of tasks, the excessive resource utilization results in increased power consumption. Furthermore, the deployment of multiple servers by cloud providers in the data centers results in the consumption of large amount of power and raises the level of CO₂ emission in the environment which is not suitable for a green cloud computing environment [13] [14] [15]. Therefore, many efforts are made to reduce power usage and emission of CO₂ through effective management of resources.

The scheduling techniques can generally be classified into static and dynamic categories based on time the user requests are generated [16]. In static scheduling, the complete information regarding the tasks and resources is known prior to execution [17]. Consequently, the decision may be performed during compilation process. The dynamic scheduling, in contrast, maps the tasks at runtime. Some scheduling techniques use heuristic methods that aim to reduce the makespan, or schedule length, which affects execution and processing times in general. Since the heuristic methods draw conclusion without large exploration, the solutions of scheduling algorithms using this approach may not attain optimality. Other scheduling techniques implement meta-heuristic algorithms that explore a large search-space in order to find an

optimal solution. Meta heuristic methods in contrast to the heuristic methods explore locally as well as globally to obtain sufficiently optimal solutions for any optimization problem. There are many meta-heuristic approaches such as Genetic Algorithm (GA) [18], Ant Colony Optimization [19], Particle Swarm Optimization (PSO)[20] and Artificial Bee Colony (ABC) [21] optimization method to optimize scheduling in the cloud environment. These scheduling strategies also aim to reduce makespan for client requests, similar to most of the heuristic strategies. Moreover, the solutions in [22] [23] aim to minimize power in cloud environments. Similarly, hybrid techniques [24] [25] incorporate heuristic and meta-heuristic approaches combined together, while attempting to improve the exploration of search space in order to determine an optimal solution. However, most of these approaches do not either aim to concurrently reduce both makespan as well as power consumption, or are unable to produce effective schedules minimizing both objectives.

In this paper, we propose Power Optimized Task Scheduling using Genetic Algorithm (POTS-GA) to schedule the tasks to VMs in the cloud environment. The proposed algorithm reduces the overall execution time of tasks and power by using fitness function that considers both objectives to search for optimal mapping of tasks to VMs. We use genetic algorithm for task scheduling while searching a large solution space. Our experiments performed with a large number of tasks and VMs demonstrate that the proposed POTS-GA algorithm outperforms other scheduling techniques in terms of makespan and power consumption.

The remaining paper is organized as follows. Section 2 describes a brief overview of the work related to task scheduling. The proposed POTS-GA algorithm is presented in Section 3. The experimental setup with system configuration and the results of evaluation are discussed in Section 4. The paper is concluded with findings and future directions in Section 5.

2. Related Work

Aimed at optimizing diverse objectives, several task scheduling strategies have recently been proposed by the researchers. Table 1 and Table 2 show a comparative analysis of the prominent research work conducted for heuristic and meta-heuristic based scheduling strategies, respectively. The comparison is performed in terms of major contribution, features and weaknesses. The scheduling strategies deploying heuristic, meta-heuristic or hybrid approaches for scheduling are summarized below.

Table 1. Comparison of prominent heuristic-based scheduling methodologies

Reference	Algorithm	Contribution	Main Features	Weakness
[26]	HEFT	It works on reduction of execution time (Makespan)	Use combination of two algorithms which are HEFT and critical path on processor (CPOP)	There is no evaluation of power consumption.
[27]	SHEFT	Improves Makespan	A directed cyclic graph(DAG) is used for weights calculation	Although there has been improvement in resource utilization but power consumption is not considered.
[28]	Min-Min Max Min	It improves makespan as well as resource utilization for small tasks	It implements combination of two scheduling approaches according to size of tasks	It is evaluated on small data in grid environment. without any consideration of power consumed.
[29]	TSACS	Focus on improving schedule length and load balancing.	Travelling salesman approach (TSP) is used for the selection of tasks.	There is no consideration of power consumption for data centers.
[30]	FCFS, RRA	Minimize makespan	The approach maps the tasks to VMs according to their arrival time in a queue	It shows better results in start and its performance is ceased with the passage of time. Not suitable for multiple objectives.

Table2. Comparison of prominent meta heuristic-based scheduling methodologies

Reference	Algorithm	Contribution	Main Features	Weakness
[22]	TS-GA	Reduce makespan and cost of tasks	Tournament Selection is used in Genetic Algorithm (GA) for the	The performance of the algorithm is evaluated on small dataset with no

			selection of tasks	consideration of power consumed.
[31]	MGGS	Improves execution time and resource utilization	It uses Greedy algorithm for Updation of vectors and roulette wheel is used for the selection method.	It combines GA with greedy strategy, while the power consumption is not optimized.
[18]	ETA-GA	It reduces overall completion time of tasks and chances of failure	It has multi-objective optimization.	It converges early and consequently, the performance of algorithm drops as the number of cloudlets increases.
[32]	Multi-objective PSO	Optimize task execution time and cost	A set of dominated and non-dominated particles is kept.	For optimization, the approach does not include power consumption.

In heuristic-based scheduling algorithms, the Heterogeneous-Earliest-Finish-Time (HEFT) algorithm proposed by Topcuoglu et al. [26] is a widely used algorithm for scheduling tasks on heterogeneous systems. The tasks at each stage are selected rank-wise and assigned to the processors, while employing an insertion-based strategy to minimize the task's earliest finish time. Lin et al. [27] proposed the scalable HEFT algorithm to allocate the workflow dynamically on cloud platforms by choosing a resource that has the shortest completion time from the current free resources. The resources that remain idle from a given threshold are released elastically at run time. Etminani et al. [28] present an algorithm to improve resource utilization and makespan for grid environment. The proposed method combines the Min-Min and Max-Min scheduling algorithms in order to gain the advantages of both algorithms. The Max-Min approach is executed for large sized tasks in the queue, whereas, Min-Min algorithm works well for a large number of small sized tasks. Similarly, Gupta et al. [33] propose several variants of HEFT based on consideration of communication cost. The proposed approach is shown to improve performance of HEFT, however, the variants work with higher complexity than HEFT and are applicable to workflows having dependencies among tasks.

A Trust Aware Distributed and Collaborative Scheduler (TADCS) is proposed by Rouzaud et al. [34] that allocates the tasks on heterogeneous VMs based on the user objectives, trust and protection to optimize the power as well as to increase the performance. Elzeki et al. [35] use max-min scheduling approach in which execution time is used for the selection of tasks instead of completion time to reduce both execution and waiting time. The task with the highest execution time is mapped to resource producing the lowest completion time to improve overall response time.

An integer programming based optimization technique is presented by Liu et al. [36] with the aim to reduce energy consumption and overall processing time of the tasks. Their method sorts all servers based on energy consumption of resources, and then assigns jobs in a greedy manner to the most energy-efficient server to optimize power. Buyya et al. [37] focus on developing dynamic resource provisioning and provide methods for efficiently managing workloads between datacenters to optimize energy conservation and enhance the performance of data centers. Their algorithm explains the VM allocation policies that consider both QoS and power consumption.

An energy-efficient task scheduling algorithm (ETSA) was presented by Panda et al. [38] for scheduling heterogeneous workloads in cloud environment. The ETSA algorithm makes a scheduling decision after considering the task's completion time and overall resource utilization. The task with minimum normalization value is mapped to virtual machine to reduce makespan as well as power consumption. Similarly, a task scheduler based on travelling salesman problem (TSP) is proposed by Nasr et al. [29] for task scheduling to improve makespan. In the proposed algorithm, tasks are grouped as clusters, then execution time of each cluster is calculated to create a matrix similar to the TSP matrix, and at the end by using the nearest neighbor algorithm clusters are mapped to virtual machines.

The Cloud Acknowledgement Scheme (CAACKS), a new method for acknowledging packets by using a single hop cloud in order to reduce energy consumption and improve network performance is proposed by Kaja et al. [39]. The proposed approach uses variable time and variable frequency based algorithms to minimize energy and increase performance. The shortest-round-vibrant-queue (SRVQ) algorithm proposed by Jeevitha et al. [40] combines the shortest-job-first algorithm and the round-robin method, while optimizing the energy consumption. All tasks are sorted in ascending order by burst time, which is subsequently used to optimize makespan. The method employs the voltage & frequency scaling approach, which aims to minimize process waiting time and enhance the efficiency of energy consumption.

In [30], the author describes First-Come-First-Serve (FCFS), Round-Robin, Shortest-Job-First, Min-Min, and Max-Min scheduling algorithms in detail. The FCFS approach maps the cloudlets to virtual machines according to their arrival time in a queue, and the allocation policy of algorithm is simply sequential. The Round-Robin algorithm iteratively allocates virtual machines to tasks with each iteration starting from first virtual machine and allocation of task is decided by time slice. In Min-Min task scheduling algorithm, small tasks are mapped first, while in Max-Min scheduling, the large tasks are executed first.

Li et al. [41] propose a greedy approach to decrease the overall completion time. In their scheduling approach, the tasks are initially arranged in descending order w.r.t their length while virtual machines are arranged in descending order according to processing power. After sorting, the tasks are iteratively mapped to virtual machines like RRA.

In terms of the meta-heuristic-based scheduling strategies, Rekha et al. [18] proposed the ETA-GA scheduling approach to perform mapping of tasks to resources based on resource capability. The task completion time and failure probability are used to compute the fitness of each chromosome. The ETA-GA approach chooses two chromosomes from the population for crossover and mutation based on fitness value. The algorithm suffers from pre-mature convergence due to selection of best chromosome in each generation. Moreover, the power consumption is not optimized, which results in VMs consuming more energy. Similarly, Zhao et al. [42] suggest a GA-based optimization method for allocating independent tasks to dynamic resources in cloud computing environments. The algorithm encodes chromosomes as digital strings, while fitness function takes into account the deadline criteria of task completion for optimizing resource usage and execution time. Soulegan et al.[43] propose another genetic algorithm based approach for task scheduling. Their approach uses fitness function as a sum of cost and makespan for optimizing the schedules. Similarly, a power-aware approach using non-dominated sorting genetic algorithm for scheduling on cloud platforms is given by Khan [11]. The approach initially optimizes the objective functions and then arranges VM indexes so that the VMs with higher weights are considered for frequent assignment to tasks.

Kaur et al. [23] combine two heuristics corresponding to assigning of the longest cloudlet and the shortest cloudlet to the fastest processor for task scheduling in cloud environment while taking into account the parameters of computational complexity and the capacity of resources. The characteristics of heuristics and randomization are also incorporated in their strategy to increase population diversity and find a better solution with a short makespan. Similarly, the Tournament Selection based Genetic Algorithm (TS-GA) is proposed by

Hamad et al. [22]. Their algorithm uses binary encoding scheme for population initialization. Its objective function considers overall finish time of tasks on available resources. When compared to round-robin and simple genetic algorithms, TS-GA performs better for small number of VMs.

Ying et al.[12] introduce an energy-aware task scheduling algorithm to improve the makespan and energy consumption using GA. The Dynamic Voltage Scaling (DVS) is employed to allow CPUs to react dynamically on various voltage supply levels to increase energy efficiency. The entire search space is explored by implementing a genetic algorithm. A modified genetic algorithm with enhanced max-min algorithm is introduced by Singh et al. [44] for scheduling independent tasks. The enhanced max-min algorithm is used to generate initial population to get optimal results for makespan. The largest task based on execution time is assigned to the VM having small amount of processing power. Similarly, in [45] [46] the authors also use genetic algorithm to reduce response time and energy consumption. In hybrid scheduling approaches, Alsaidy et al. [25] suggest heuristics for initialization of solutions for PSO. The heuristics in the hybrid approach allocate VMs to tasks using minimum execution time of a task computed among all virtual machines and a round-robin assignment mechanism. Another hybrid technique [24] combining heuristic and metaheuristic strategies also attempts to improve the exploration of search space in order to determine an optimal solution by incorporating the Shortest-Job-to-Fastest-Processor (SJFP) method along with PSO. The authors introduce SJFP heuristic in initialization phase for a better selection of overall population. Similarly, Manasrah et al. [47] propose a hybrid algorithm that divides iterations in PSO and GA algorithms. The population is initially updated using the iterations of GA algorithm. Subsequently, the population generated by the GA algorithm is used by PSO for optimizing schedules of workflows executing on cloud platforms.

In contrast to the methodologies stated above, this paper proposes power-optimized task scheduling using genetic algorithm (POTS-GA) that aims to efficiently execute user requests to generate small schedule length, while optimizing makespan as well as power consumption. The finish time of each VM and overall power consumption of tasks on virtual machines are used for computing fitness and subsequent assignment of VMs to tasks. Through consideration of both objectives, the POTS-GA strategy is able to significantly perform better than other approaches in terms of makespan and power consumption.

3. Power Optimized Task Scheduling using Genetic Algorithm (POTS-GA)

The POTS-GA scheduling approach uses genetic algorithm to search for solution space in order to get optimal scheduling. It aims at minimizing power consumption as well as reducing response time without affecting task performance.

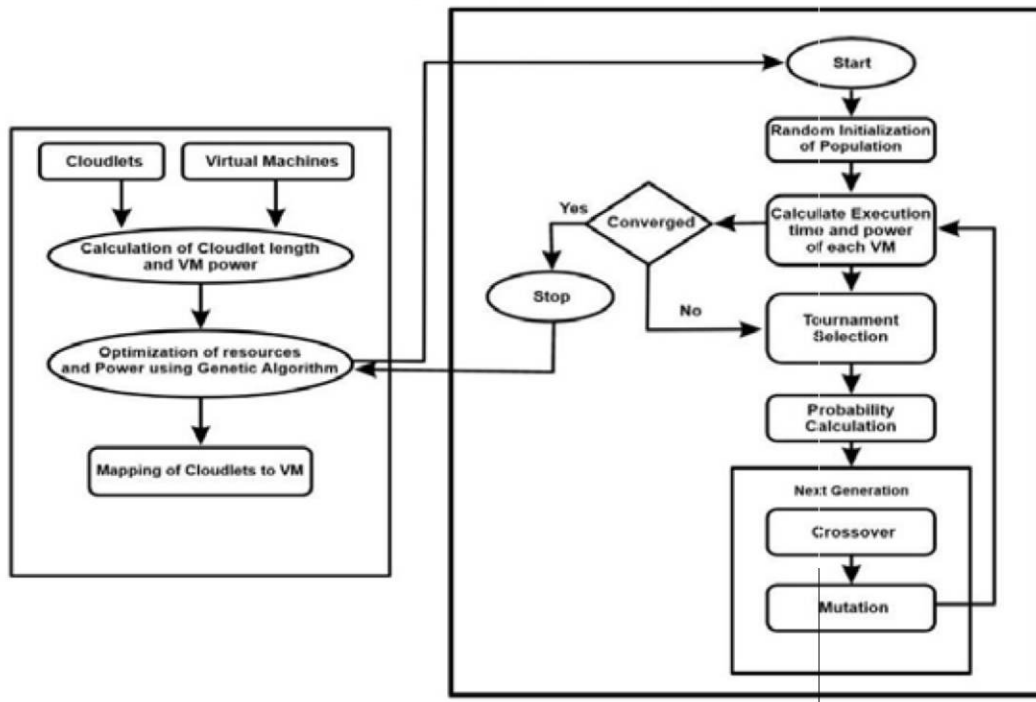


Figure 1. POTS-GA Scheduler

Figure 1 depicts the main phases of the proposed POTS-GA algorithm. The tasks of various sizes and virtual machines with diverse processing capability and power consumption are used as input by the algorithm to get best mapping of tasks to VMs. The GA starts with a random population of individual chromosomes, where each chromosome represents a feasible mapping. The fitness of each individual chromosome is assessed by using the overall execution time and the power of virtual machines. The genetic operators i.e. crossover and mutation are used to generate new viable solutions. The chromosomes (mappings) with low makespan and power are likely to be retained among next generations. The tasks are then mapped to virtual machines using the most appropriate mapping found through the chromosome having the best fitness value.

The major steps of the POTS-GA scheduling approach are detailed below:

3.1. Problem Encoding and Population Initialization

The POTS-GA algorithm uses a discrete encoding scheme for representing a chromosome as a collection of $1 \times n$ values, where n represents the number of tasks. The tasks are to be mapped to m virtual machines. The entire population of chromosomes is initialized with chromosomes having random gene values, where each gene value represents index of the VM to be considered for allocation to a task.

Assume the tasks $T_1, T_2, T_3, \dots, T_{10}$ to be mapped to virtual machines V_1, V_2, V_3, V_4, V_5 , where more than one task can be mapped to a VM. A chromosome with values $\{3, 5, 3, 2, 5, 1, 3, 2, 3, 2\}$ would imply the task T_1 being mapped to V_3 . Similarly, $T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}$ are mapped to $V_5, V_3, V_2, V_5, V_1, V_3, V_2, V_3, V_2$, respectively. The POTS-GA algorithm searches for the mapping with the objective of minimizing makespan and power consumption.

3.2. Fitness Function

The fitness of a chromosome in the POTS-GA algorithm depends upon both the makespan and the power consumption. The computation of makespan requires execution time and finish time of the virtual machines, whereas, the power consumption is based on the power requirement of the machine and the execution time of tasks. For our scenario, we formulate the problem as $Min(\Theta(\psi, W))$, as discussed below.

The execution time $E_{(T_i, V_k)}$ of a task T_i that is assigned to the VM V_k is computed as:

$$E_{(T_i, V_k)} = \frac{T_i.length}{V_k.mips} \quad (1)$$

where $T_i, \forall i=0, 1, 2, \dots, n-1$, represents tasks, the variable *length* represents the individual length or number of instructions of each task, and *mips* is the processing capability of each virtual machine in millions of instructions per second.

The finish time of a virtual machine V_k is the sum of execution times of tasks $T_s \subseteq T$, assigned to the virtual machine, as given below:

$$F_{V_k} = \sum_{i=0}^{|T_s|-1} E_{(T_{s_i}, V_k)}, \quad (2)$$

where $F_{V_k}, k=0, 1, 2, \dots, m-1$ represents the finish time of tasks T_s assigned to each virtual machine V_k . Since the virtual machines are running concurrently, the makespan ψ is the maximum finish time among all VMs.

$$\psi = \max(F_{V_k}), \forall k=0,1,2,\dots,m-1 \quad (3)$$

The power consumption W for overall execution of tasks is computed as:

$$W = \psi * \left(\frac{pwr}{3600*1000}\right) \quad (4)$$

where pwr represents the power consumption (in kWh) of the host on which the virtual machines are running.

The fitness of a chromosome is computed as the sum of scaled values of makespan and power consumption. Let ψ^{max} & ψ^{min} represent the maximum and minimum makespan values, and W^{max} & W^{min} represent the maximum and minimum power consumption among all mappings (chromosomes,) as computed below:

$$\psi^{max} = n * \left(\frac{(\max(\sum_{i=0}^{n-1} C_i.length))}{(\min(\sum_{i=0}^{m-1} V_i.mips))}\right) \quad (5)$$

$$\psi^{min} = \left(\frac{(\min(\sum_{i=0}^{n-1} C_i.length))}{(\max(\sum_{i=0}^{m-1} V_i.mips))}\right) \quad (6)$$

$$W^{max} = \psi^{max} * \left(\frac{pwr}{3600*1000}\right) \quad (7)$$

$$W^{min} = \psi^{min} * \left(\frac{pwr}{3600*1000}\right) \quad (8)$$

For a chromosome, having power W and makespan ψ , the fitness θ is calculated as:

$$\theta = \frac{\psi - \psi^{min}}{\psi^{max} - \psi^{min}} + \frac{W - W^{min}}{W^{max} - W^{min}} \quad (9)$$

The two objectives of minimizing the power consumption and makespan are integrated into the fitness value θ that is used by the POTS-GA algorithm to search for optimal mapping.

3.3. Selection Operator

The selection operator used in the proposed algorithm is tournament selection which selects k -individuals from the population. The fitness of each chromosome based on execution time and power is calculated and the selected ' k ' chromosomes are sorted in order of fitness values. The best chromosomes with high fitness value are then selected for generating new chromosomes through crossover and mutation.

3.4. Crossover

The POTS-GA algorithm uses single-point & two-point crossover operators to produce offspring for a new population. For instance, two chromosomes (having indices of mapped virtual machines) with a randomly selected crossover point $C=5$ for one-point crossover, and the corresponding output chromosomes obtained after crossover are shown in Figures 2 and 3, respectively.

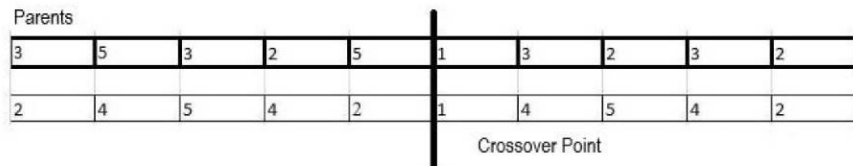


Figure 2. Chromosomes before Single Point Crossover

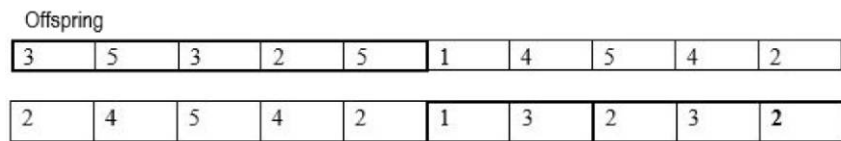


Figure 3. Chromosomes after Single-Point Crossover

Similarly, for the two-point crossover with two randomly chosen crossover points $C_1=4$ and $C_2=8$, the input chromosomes and the offspring are shown in Figures 4 and 5, respectively.

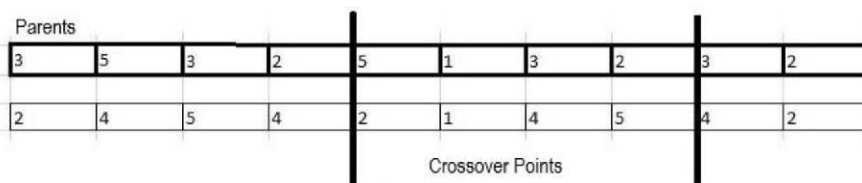


Figure 4. Chromosomes before the Two-Point Crossover

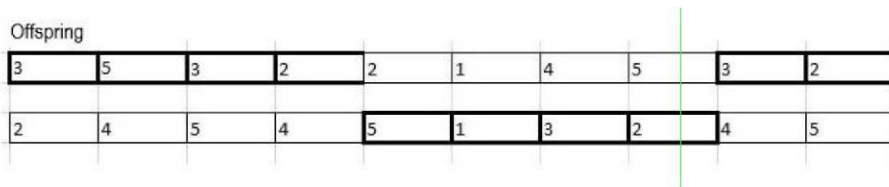


Figure 5. Chromosomes after Two-Point Crossover

3.5. Mutation

The POTS-GA algorithm performs mutation (Figure 6) to generate random VM indices to be used for mapping. The mutation rate with a probability defines the frequency of replacing VM indices with new indices.

Before Mutation									
3	5	3	2	5	1	4	5	4	2
After Mutation									
2	4	4	5	5	1	5	4	3	2

Figure 6. Mutation Operation

For mutation, a random integer r , $0 \leq r \leq 1$ is generated and the VM indices in a chromosome are replaced only if the value of r exists within the range of the mutation probability.

3.6. Find Best Mapping

The POTS-GA algorithm uses the fitness value to determine the best mapping. The chromosome producing the lowest fitness value is computed and compared with the previous best chromosome at each iteration for update.

3.7. Allocation of Virtual Machines

After execution of all iterations, the chromosome having the best fitness value is used for assigning tasks to VMs. The individual genes representing VMs are allocated to the corresponding tasks.

Algorithm 1: Power optimized task scheduling using genetic algorithm (POTS-GA)

- 1: /* Let $V_j, \forall j = 0, 1, 2, \dots, m-1$ represent the set of virtual machines, each having different parameters and let $T_i, \forall i = 0, 1, 2, \dots, n-1$ be the set of tasks, each being characterized with different features. The variable η is used to represent the tournament size. Let $P_i, \forall i = 0, 1, 2, \dots, |P|-1$ represent the population of chromosomes */
- 2: **Begin**
- 3: $x = 1, Max = 1000$
- 4: // Main loop to search for best mapping
- 5: **while** ($x \leq Max$) **do**

```

6:      //Initialize the population  $P$  chromosomes with random
      //VM indices  $R_j$ 
       $P_i = \{R_j, \forall j=0,1,2,\dots,n-1\}, \forall i=0,1,2,\dots, |P|-1$ 
7:      if ( $x == 1$ ) then
8:           $B=0$  //Initialize index for the global best
              //chromosome
9:      end if
10:     Compute fitness  $\theta_i$  of each chromosome  $P_i, \forall i=0,1,2,\dots,|P|-1$  by
      using Equations (1-9) of Section 3.2
11:      $Q=\{\}, R=\{\}, S=\{\}$  // Initialize  $Q$  and  $R$  and  $S$  as //temporary
      population variables
12:     //Select  $\eta$  chromosomes using tournament selection given in
      Section 3.3.
13:      $R = \text{tournamentSelection}(P)$ 
14:     //Let  $S$  be the remaining individuals to be passed to //next
      generation as elites
15:     if  $\eta \% 2 \neq 0$  then
16:         Add chromosome  $R_{\eta-1}$  to the population  $Q$ 
17:     end if
18:     //Apply crossover as mentioned in Section 3.4.
19:     for  $j= 0$  to  $\eta -1$  step 2 do
20:          $(u,v) = \text{crossover}(P, j, j+1)$  // return pair of
              //chromosomes  $u$  and  $v$ 
21:         Add chromosomes  $u$  and  $v$  to population  $Q$ 
22:     end for
23:     //Apply mutation operator as described in Section 3.5
24:      $P = \text{mutation}(Q)$ 
25:     Add  $S$  to population  $P$ 
26:      $x = x+1$ 
27:     //Find index of the chromosome having the best fitness
      //values as given in Section 3.6
28:      $k = \text{findBestMapping}()$ 
29:     if  $\theta_B < \theta_k$  then

```

```

30:           B = k //Update the global best chromosome
31:       endif
32:   end while
33:   //Allocate the tasks to VMs
34:   for  $j=0,1,2,\dots,n-1$  do
35:       Map task  $T_j$  to the VM  $P_B[j]$ 
36:   end for
37: End

```

The POTS-GA algorithm (Algorithm 1) performs initialization of the variables for iteration count and maximum iteration at step 3. At step 5, the loop is iterated for the given iteration count to search for optimal mapping. The step 6 initializes the population with random virtual machine and index of the global best chromosome **B** is initialized at steps 7-9. The fitness of each individual chromosome based on makespan and power consumption is computed at step 10. The step 11 initializes temporary population variables. The η chromosomes are selected through tournament selection at step 13. The steps 15-22 add chromosomes to the population **Q** in the form of pairs. The remaining chromosome is added using steps 15-17, while the steps 19-22 perform crossover to add new chromosomes as pairs to the population **Q**. The mutation operator is performed on the population **Q** to produce the new population **P** at step 24. The remaining chromosomes **S** are added to the population **P**, and the loop count is incremented subsequently. The steps 28-31 find the best mapping and update the global best chromosome **B**. The allocation of virtual machines to task is performed through final steps by using the elements of the best chromosome **P_B**. The complexity of the POTS-GA algorithm is $O(|P| * Max * n)$, where $|P|$, *Max*, and n represent respectively the population size, the number of iterations and the number of tasks. Since POTS-GA uses a meta-heuristic approach, its cost for large-scale environments may be reduced by limiting the population size and the number of iterations.

4. Experimentation and Results

4.1 Simulation Setup

For experimentation, the CloudSim framework [48] has been used. It includes PowerDatacenter, PowerHost and PowerVm for power-based configurations for Datacenters, Hosts and VMs, respectively. Each task contains a user request that is comprised of a number of instructions. The algorithms have been evaluated with number of tasks set to vary from 50

to 500 and the number of VMs set to vary from 4 to 16. We have performed simulation using configuration parameters given in Table 3.

Table 3. Platform and configuration parameters

Simulation Platform	CloudSim 4.0	Number of Host Machines	4
Virtual Machine Monitor (VMM)	Xen	Task Length (Millions of instructions)	1000-2000
Task Scheduler	Space Shared	VM processing elements	1
System Architecture	X86	Number of Tasks	50,100,...,500
Processing elements of Tasks	1	Number of VMs	4,8,16
Number of Datacenters	2	VM RAM	512MB
VM Bandwidth	10 Megabits/s	Power (Watts)	200-300

Table 4. Parameters used for the POTS-GA algorithm

Encoding	Discrete	Crossover probability	0.8
Population Size	100	Mutation operator	Random resetting
Max iterations	1000	Mutation probability	0.05
Crossover operator	30% Single point and 70% two- point	Tournament size	80% of population

The parameters used for the POTS-GA algorithm are given in Table 4. For performance evaluation, the POTS-GA algorithm is compared with other algorithms including the ETA-GA , Robin algorithm (RRA), FCFS, and Greedy algorithm. The algorithms are evaluated in

terms of makespan and power consumption. The makespan or schedule length is the total amount of time required for execution of the tasks mapped to virtual machines. Similarly, the power consumption (kWh) represents the power consumed by the virtual machines for executing all the scheduled tasks.

4.2 Evaluation Outcomes

This section provides an overview of the performance results of power and makespan for the 04, 08, and 16 virtual machines, respectively.

4.2.1 Makespan

Figure 7 depicts the makespan of various methods utilizing four VMs with diverse numbers of tasks, ranging from 50 to 500. The performance of the POTS-GA algorithm and the ETA-GA algorithm is nearly identical when the number of tasks is small. It is evident from the figure that with a large number of tasks, the performance of ETA-GA degrades beyond the batch size of 300. Overall, POTS-GA surpasses all other strategies in terms of makespan. The ETA-GA algorithm performs next to the POTS-GA algorithm by producing small makespan values.

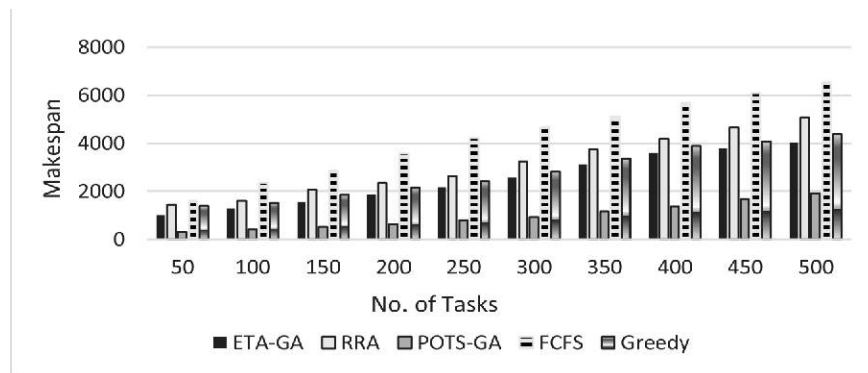


Figure 7. Makespan for different number of tasks using 04 virtual machines

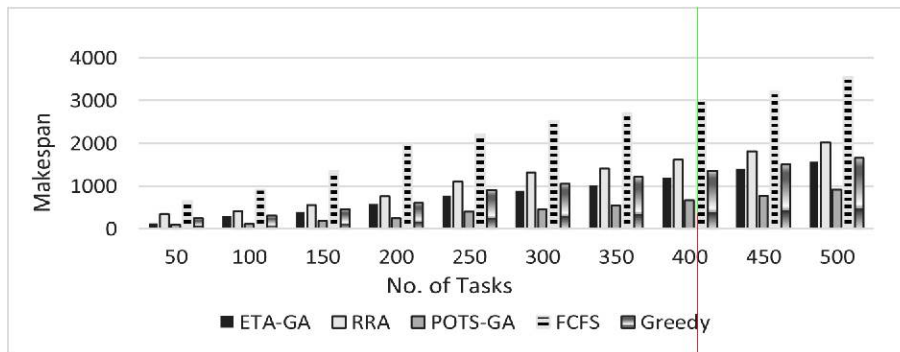


Figure 8. Makespan for different number of tasks using 08 VMs

With eight VMs, the makespan for the evaluated algorithms with various number of tasks is shown in Figure 8. The proposed technique POTS-GA has drastically produced small makespan when compared to all other benchmark scheduling techniques. When there are a few tasks, the RRA algorithm performs better than the Greedy algorithm. As the number of tasks is increased, the RRA performs the worst by producing large makespan values. Followed by the performance of the POTS-GA algorithm, the ETA-GA algorithm performs significantly better than RRA, Greedy and FCFS for all batch sizes.

The schedule length for various scheduling algorithms, with different number of tasks being executed on 16 VMs is depicted in Figure 9. Similar to previous scenarios, when virtual machines are small in number, the POTS-GA algorithm continues to outperform other task scheduling techniques in this case when the number of VMs is large.

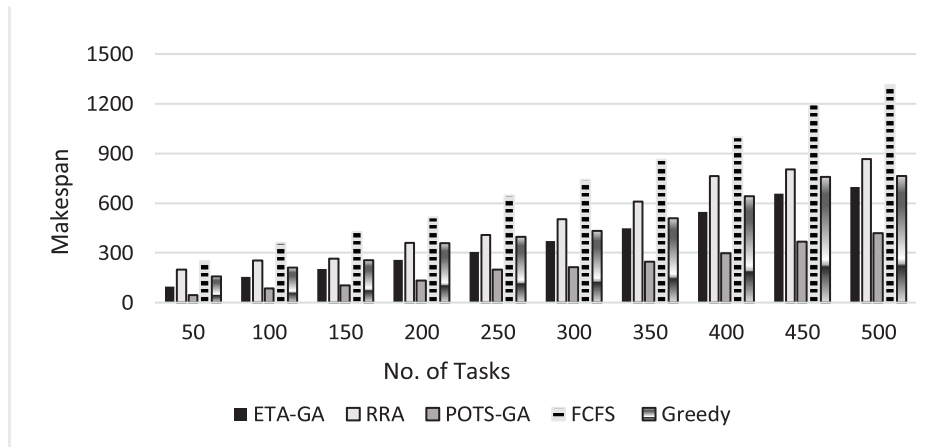


Figure 9. Makespan for different number of tasks using 16 virtual machines

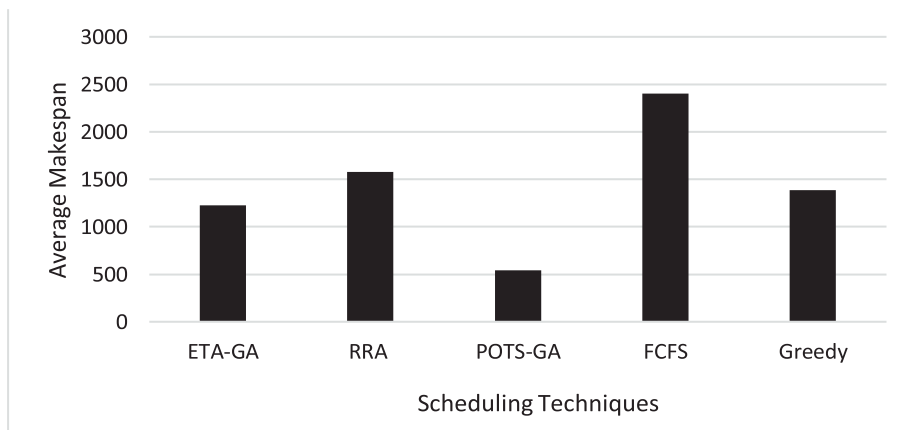


Figure 10. Average Makespan obtained for all configurations of VMs

The overall average makespan of various scheduling techniques is shown in Figure 10. The results depict that the proposed POTS-GA algorithm provides better execution time in almost all batch sizes of tasks. Overall, the POTS-GA algorithm achieves 65%, 75%, 80% and 73% average improvement over the ETA-GA, RRA, FCFS and Greedy algorithms, respectively, in terms of schedule length.

4.2.2 Power Consumption

The power consumed for execution of tasks using the schedule generated by the various scheduling strategies, with different number of tasks and VMs is given in Table 5.

Table 5. Power Consumption for different number of tasks using 04 virtual machines

	Tasks									
	50	100	150	200	250	300	350	400	450	500
ETA-GA	42.82	67.46	143.60	182.60	210.30	253.3	284.20	332.20	367.10	407.50
RRA	52.39	85.52	133.18	177.94	204.06	220.37	269.66	323.02	352.33	379.19
POTS-GA	2.97	8.42	13.92	30.51	39.94	46.52	70.03	87.12	119.86	142.26
FCFS	91.23	157.8	215.45	278.18	327.64	367.52	398.78	426.34	473.36	512.93
Greedy	62.9	92.8	110.45	158.3	181.6	210.3	267.5	305.5	335.2	375.2

The power consumption incurred for schedules produced by ETA-GA and the proposed strategy POTS-GA is similar to some extent when number of tasks is small. But the power consumption for execution of tasks scheduled through ETA-GA is exponentially increased for large number of tasks. Table 5 clearly depicts that the power consumption incurred for execution of tasks scheduled through POTS-GA is significantly less than RRA, FCFS, Greedy and ETA-GA.

Table 6. Power consumption for different number of tasks using 08 virtual machines

	Tasks									
	50	100	150	200	250	300	350	400	450	500
ETA-GA	21.86	39.98	60.45	86.29	114.46	148.54	191.89	251.97	298.25	367.72
RRA	49.88	95.32	172.50	201.86	256.00	286.44	310.73	347.12	415.25	456.68
POTS-GA	7.62	11.92	28.07	35.94	41.52	66.03	85.12	121.86	146.26	167.98

FCFS	100.82	165.24	215.2	295.72	385.87	476.21	543.06	597.94	655.14	715.47
Greedy	76.24	92.85	167.28	197.36	256.65	298.57	343.42	368.24	386.92	395.56

The power consumption of different algorithms having various number of tasks with eight VMs is shown in Table 6. Overall, the POTS-GA outperforms all other scheduling strategies including the ETA-GA algorithm. For large number of tasks, the Greedy algorithm outperforms RRA in terms of power consumption. The FCFS algorithm results in the highest power consumption for all configurations with varying number of tasks.

Table 7: Power Consumption incurred for execution of tasks on 16 VMs

	Tasks									
	50	100	150	200	250	300	350	400	450	500
ETA-GA	27.63	85.82	165.45	196.56	196.46	232.54	254.89	294.97	323.92	362.72
RRA	42.87	94.54	195.92	257.86	277.00	358.74	408.07	449.12	487.76	510.87
POTS-GA	7.41	20.92	35.07	54.94	71.52	88.03	125.12	136.86	162.26	185.87
FCFS	98.68	163.54	272.32	345.67	412.79	495.44	578.07	654.28	714.83	847.65
Greedy	52.85	103.78	207.26	244.65	256.57	297.94	315.24	388.43	405.56	432.48

Table 7 illustrates the power consumption of POTS-GA and other methods while using 16 virtual machines with varying number of tasks. According to Table 7, POTS-GA consumes less power than FCFS, RRA, Greedy, and ETA-GA. The FCFS algorithm results in the highest values of power consumption for all scenarios. The ETA-GA algorithm consumes less energy for 50 to 250 tasks. However, as the number of tasks reaches up to 300, the performance of the ETA-GA algorithm starts to deteriorate. Similar to the results for 08 VMs, the RRA algorithm generates schedules that incur less power consumption in comparison with the FCFS algorithm.

Figure 11 shows the overall power usage of several scheduling strategies with various virtual machines. As shown in the diagram that the POTS-GA consumes less power than other approaches for all batch sizes of tasks using different number of VMs. The average improvement in power consumption achieved by POTS-GA over the ETA-GA, RRA, FCFS and Greedy algorithms is 66%, 76%, 82%, and 74%, respectively.

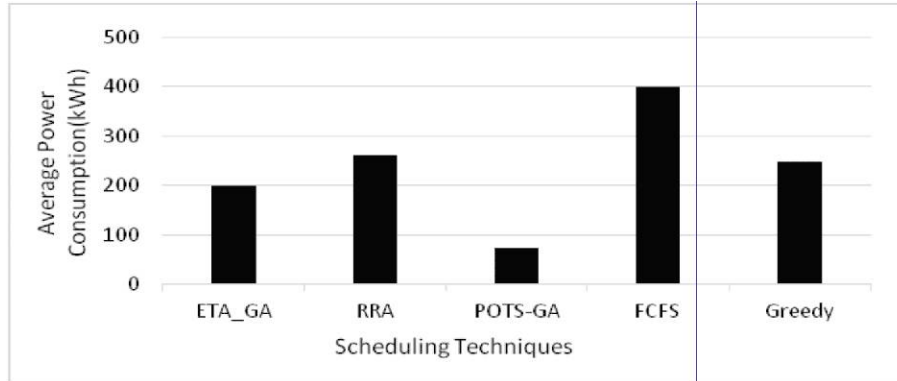


Figure 11. Average power consumption for all configurations of virtual machines

5. Conclusion

For cloud computing environments, task scheduling is considered to be a major issue due to its complexity. On high performance cloud platforms, an effective task scheduling can improve overall execution time as well as power consumption. In this paper, we have proposed the POTS-GA algorithm that aims to optimize execution time and power consumption by generating small length schedules while simultaneously considering the power of each virtual machine. The genetic algorithm uses power of virtual machine and makespan for computing fitness of each chromosome. After a number of iterations, the chromosome with best mapping is found and the tasks are then assigned to VMs. We have conducted experiments using various number of tasks and VMs to evaluate the performance of POTS-GA and other algorithms. The performance results show that the POTS-GA algorithm performs 65%, 75%, 80% and 73% better for makespan and 66%, 76%, 82%, and 74% better for power consumption than the ETA-GA, RRA, FCFS, and Greedy algorithms, respectively.

For our current implementation, the execution cost & resource utilization are not considered in our proposed approach, so we aim to extend our work by considering these optimization metrics in future. Moreover, a heuristic for initializing the population may be incorporated to enhance performance of the proposed algorithm.

Acknowledgment

This research work was accomplished as part of the research work conducted during MS Thesis at BZU Multan, Pakistan. The authors are thankful to the Department of Computer Science, BZU, Multan, for provision of equipment required to carry out this research work.

References

- 1 Zhang, S., Zhang, S., Chen, X., & Huo, X. (2010). Cloud computing research and development trend. *2nd International Conference on Future Networks, ICFN 2010*, 93–97. <https://doi.org/10.1109/ICFN.2010.58>
- 2 Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. <https://doi.org/10.1145/1721654.1721672>
- 3 Odun-Ayo, I., Ananya, M., Agono, F., & Goddy-Worlu, R. (2018). Cloud Computing Architecture: A Critical Analysis. *Proceedings of the 2018 18th International Conference on Computational Science and Its Applications, ICCSA 2018*, 1–7. <https://doi.org/10.1109/ICCSA.2018.8439638>
- 4 Birke, R., Chen, L. Y., & Smirni, E. (2012). Data centers in the cloud: A large scale performance study. *Proceedings - 2012 IEEE 5th International Conference on Cloud Computing, CLOUD 2012*, 336–343. <https://doi.org/10.1109/CLOUD.2012.87>
- 5 Yuan, H., Kuo, C. C. J., & Ahmad, I. (2010). Energy efficiency in data centers and cloud-based multimedia services: an overview and future directions. *2010 International Conference on Green Computing, Green Comp 2010*, 375–382. <https://doi.org/10.1109/GREENCOMP.2010.5598292>
- 6 Abdessamia, F., Tai, Y., Zhang, W. Z., & Shafiq, M. (2017). An improved particle swarm optimization for energy-efficiency virtual machine placement. *Proceedings - 5th International Conference on Cloud Computing Research and Innovation, ICCCRI 2017*, 7–13. <https://doi.org/10.1109/ICCCRI.2017.9>
- 7 Bobroff, N., Kochut, A., & Beaty, K. (2007). Dynamic placement of virtual machines for managing SLA violations. *10th IFIP/IEEE International Symposium on Integrated Network Management 2007, IM '07*, 5, 119–128. <https://doi.org/10.1109/INM.2007.374776>
- 8 Malekloo, M. H., Kara, N., & El Barachi, M. (2018). An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments. *Sustainable Computing: Informatics and Systems*, 17, 9–24.

<https://doi.org/10.1016/j.suscom.2018.02.001>

- 9 Pietri, I., & Sakellariou, R. (2016). Mapping virtual machines onto physical machines in cloud computing: A survey. *ACM Computing Surveys*, 49(3). <https://doi.org/10.1145/2983575>
- 10 Al-Maytami, B. A., Fan, P., Hussain, A., Baker, T., & Liatsist, P. (2019). A Task Scheduling Algorithm with Improved Makespan Based on Prediction of Tasks Computation Time algorithm for Cloud Computing. *IEEE Access*, 7, 160916–160926. <https://doi.org/10.1109/ACCESS.2019.2948704>
- 11 Khan, M. A. (2022). A cost-effective power-aware approach for scheduling cloudlets in cloud computing environments. *Journal of Supercomputing*, 78(1), 471–496. <https://doi.org/10.1007/s11227-021-03894-2>
- 12 Ying, C. T., & Yu, J. (2012). Energy-aware genetic algorithms for task scheduling in cloud computing. *Proceedings - 7th ChinaGrid Annual Conference, ChinaGrid 2012*, 43–48. <https://doi.org/10.1109/ChinaGrid.2012.15>
- 13 Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Yousafzai, A., & Xia, F. (2015). A survey on virtual machine migration and server consolidation frameworks for cloud data centers. *Journal of Network and Computer Applications*, 52, 11–25. <https://doi.org/10.1016/j.jnca.2015.02.002>
- 14 Duy, T. V. T., Sato, Y., & Inoguchi, Y. (2010). Performance evaluation of a green scheduling algorithm for energy savings in cloud computing. *Proceedings of the 2010 IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum, IPDPSW 2010*, 1–8. <https://doi.org/10.1109/IPDPSW.2010.5470908>
- 15 Jafarnejad Ghomi, E., Masoud Rahmani, A., & Nasih Qader, N. (2017). Load-balancing algorithms in cloud computing: A survey. *Journal of Network and Computer Applications*, 88(December 2016), 50–71. <https://doi.org/10.1016/j.jnca.2017.04.007>
- 16 Deepa, T., & Cheelu, D. (2018). A comparative study of static and dynamic load balancing algorithms in cloud computing. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017*, 3375–3378. <https://doi.org/10.1109/ICECDS.2017.8390086>
- 17 Manglani, V., Jain, A., & Prasad, V. (2018). Task Scheduling in Cloud Computing

- Environment. *International Journal of Computer Sciences and Engineering*, 6(5), 513–515. <https://doi.org/10.26438/ijcse/v6i5.513515>
- 18 Rekha, P. M., & Dakshayini, M. (2019). Efficient task allocation approach using genetic algorithm for cloud environment. *Cluster Computing*, 22(4), 1241–1251. <https://doi.org/10.1007/s10586-019-02909-1>
- 19 Wang, Y., Zuo, X., Wu, Z., Wang, H., & Zhao, X. (2022). Variable neighborhood search based multiobjective ACO-list scheduling for cloud workflows. *Journal of Supercomputing*, 78(17), 18856–18886. <https://doi.org/10.1007/s11227-022-04616-y>
- 20 Nabi, S., Ahmad, M., Ibrahim, M., & Hamam, H. (2022). AdPSO: Adaptive PSO-Based Task Scheduling Approach for Cloud Computing. *Sensors*, 22(3), 1–22. <https://doi.org/10.3390/s22030920>
- 21 Navimipour, N. J. (2015). Task scheduling in the Cloud Environments based on an Artificial Bee Colony Algorithm. *Proceedings of 2015 International Conference on Image Processing, Production and Computer Science (ICIPCS'2015)*, 38–44.
- 22 Hamad, S. A., & Omara, F. A. (2016). Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment. *Computer Science and Application*, 06(06), 317–322. <https://doi.org/10.12677/csa.2016.66038>
- 23 Kaur, S., & Verma, A. (2012). An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment. *International Journal of Information Technology and Computer Science*, 4(10), 74–79. <https://doi.org/10.5815/ijitcs.2012.10.09>
- 24 Abdi, S., Motamedi, S. A., & Sharifian, S. (2014). Task Scheduling using Modified PSO Algorithm in Cloud Computing Environment. *Proceedings - International Conference on Machine Learning, Electrical and Mechanical Engineering, ICMLEME 2014*, 37–41. <https://dx.doi.org/10.15242/IIE.E0114078>
- 25 Al-Saidy, S. A., Abbood, A. D., & Sahib, M. A. (2022). Heuristic initialization of PSO task scheduling algorithm in cloud computing. *Journal of King Saud University - Computer and Information Sciences*, 34(6), 2370–2382. <https://doi.org/10.1016/j.jksuci.2020.11.002>
- 26 Topcuoglu, H., Hariri, S., & Society, I. C. (2002). *Performance-Effective and Low-*

Complexity. 13(3), 260–274.

- 27 Lin, C., & Lu, S. (2011). Scheduling scientific workflows elastically for cloud computing. *Proceedings - 2011 IEEE 4th International Conference on Cloud Computing, CLOUD 2011*, 746–747. <https://doi.org/10.1109/CLOUD.2011.110>
- 28 Etminani, K., & Naghibzadeh, M. (2007). A min-min max-min selective algorithm for grid task scheduling. *2007 3rd IEEE/IFIP International Conference in Central Asia on Internet, ICI 2007*. <https://doi.org/10.1109/canet.2007.4401694>
- 29 Nasr, A. A., El-Bahnasawy, N. A., Attiya, G., & El-Sayed, A. (2019). Using the TSP Solution Strategy for Cloudlet Scheduling in Cloud Computing. *Journal of Network and Systems Management*, 27(2), 366–387. <https://doi.org/10.1007/s10922-018-9469-9>
- 30 Salot, P. (2016). A survey of scheduling algorithm in cloud computing environment. *International Journal of Control Theory and Applications*, 9(36), 137–145.
- 31 Zhou, Z., Li, F., Zhu, H., Xie, H., Abawajay, J.H., & Chowdhury, M.U.(2020). An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments. *Neural Computing & Application* **32**, 1531–1541. <https://doi.org/10.1007/s00521-019-04119-7>
- 32 Ramezani, F., Lu, J., Hussain, F. (2013). Task Scheduling Optimization in Cloud Computing Applying Multi-Objective Particle Swarm Optimization. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds) *Service-Oriented Computing. ICSOC 2013. Lecture Notes in Computer Science*, vol 8274. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-45005-1_17-
- 33 Gupta, S., Iyer, S., Agarwal, G., Manoharan, P., Algarni, A. D., Aldehim, G., & Raahemifar, K. (2022). Efficient Prioritization and Processor Selection Schemes for HEFT Algorithm: A Makespan Optimizer for Task Scheduling in Cloud Environment. *Electronics (Switzerland)*, 11(16). <https://doi.org/10.3390/electronics11162557>
- 34 0020Rouzaud-cornabas, J. (2011). *A Trust Aware Distributed and Collaborative Scheduler for Virtual Machines in Cloud*.
- 35 Elzeki, M.O., Z. Reshad, M., & A. Elsoud, M. (2012). Improved Max-Min Algorithm in Cloud Computing. *International Journal of Computer Applications*, 50(12), 22–27. <https://doi.org/10.5120/7823-1009>

- 36 Liu, N., Dong, Z., & Rojas-Cessa, R. (2012). Task and server assignment for reduction of energy consumption in datacenters. *Proceedings - IEEE 11th International Symposium on Network Computing and Applications, NCA 2012*, 171–174. <https://doi.org/10.1109/NCA.2012.42>
- 37 Buyya, R., Beloglazov, A., & Abawajy, J. (2010). *Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges*. May 2016. <http://arxiv.org/abs/1006.0308>
- 38 Panda, S. K., & Jana, P. K. (2019). An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems. *Cluster Computing*, 22(2), 509–527. <https://doi.org/10.1007/s10586-018-2858-8>
- 39 Kaja, S., Shakshuki, E. M., Guntuka, S., Ul, A., Yasar, H., & Malik, H. (2020). Acknowledgment scheme using cloud for node networks with energy - aware hybrid scheduling strategy. *Journal of Ambient Intelligence and Humanized Computing*, 11(10), 3947–3962. <https://doi.org/10.1007/s12652-019-01629-z>
- 40 Jeevitha, J. K., & Athisha, G. (2021). A novel scheduling approach to improve the energy efficiency in cloud computing data centers. *Journal of Ambient Intelligence and Humanized Computing*, 12(6), 6639–6649. <https://doi.org/10.1007/s12652-020-02283-6>
- 41 Li, J., Feng, L., & Fang, S. (2014). An Greedy-Based Job Scheduling Algorithm in Cloud Computing. *Journal of Software*, 9(4), 921–925. <https://doi.org/10.4304/jsw.9.4.921-925>
- 42 Zhao, C., Zhang, S., Liu, Q., Xie, J., & Hu, J. (2009). Independent task scheduling based on genetic algorithm in cloud computing. *5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, China, 2009*, pp. 1-4, doi: 10.1109/WICOM.2009.5301850.
- 43 Soulegan, N. S., Barekatin, B., & Neysiani, B. S. (2021). MTC: Minimizing Time and Cost of Cloud Task Scheduling based on Customers and Providers Needs using Genetic Algorithm. *International Journal of Intelligent Systems and Applications*, 13(2), 38–51. <https://doi.org/10.5815/ijisa.2021.02.03>
- 44 Singh, S., & Kalra, M. (2014). Scheduling of independent tasks in cloud computing

using modified genetic algorithm. *Proceedings - 2014 6th International Conference on Computational Intelligence and Communication Networks, CICN 2014*, 565–569. <https://doi.org/10.1109/CICN.2014.128>

- 45 Gabaldon, E., Lerida, J. L., Guirado, F., & Planes, J. (2017). Blacklist multi-objective genetic algorithm for energy saving in heterogeneous environments. *Journal of Supercomputing*, 73(1), 354–369. <https://doi.org/10.1007/s11227-016-1866-9>
- 46 Kar, I., Parida, R. N. R., & Das, H. (2016). Energy aware scheduling using genetic algorithm in cloud data centers. *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, 3545–3550. <https://doi.org/10.1109/ICEEOT.2016.7755364>
- 47 Manasrah, A. M. & Ali, H. B. (2018). Workflow Scheduling Using Hybrid GA-PSO Algorithm in Cloud Computing. *Wireless Communications and Mobile Computing*, 2018, 1530-8669. <https://doi.org/10.1155/2018/1934784>
- 48 Calheiros, R., Ranjan, R., Beloglazov, A., De Rose, C. A. F., & Buyya, R. (2009). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software - Practice and Experience*, 39(7), 701–736. <https://doi.org/10.1002/spe>

Accurate Attack Detection in Intrusion Detection System for cyber Threat Intelligence Feeds using Machine Learning Techniques

Ehtsham Irshad^{a*}, Abdul Basit Siddiqui^a

^a Department of Computer Science, Capital University of Science and Technology, Islamabad, Pakistan
ehtsham_irshad@hotmail.com, abasit.siddiqui@cust.edu.pk

*Corresponding Author: Ehtsham Irshad ehtsham_irshad@hotmail.com

Abstract

With the advancement of modern technology, cyber-attacks are continuously rising. Malicious behavior in the network is discovered using security devices like intrusion detection systems (IDS), firewalls, and antimalware systems. To defend organizations, procedures for detecting threats more correctly and precisely must be defined. The proposed study investigates the significance of cyber-threat intelligence (CTI) feeds in accurate IDS detection. The NSL-KDD and CSE-CICIDS-2018 datasets were analyzed in this study. This research makes use of normalization, transformation, and feature selection algorithms. Machine learning (ML) techniques were employed to determine if the traffic was normal or an attack. With the proposed study the ability to identify network attacks has improved using machine learning algorithms. The proposed model provides 98% accuracy, 97% precision, and 96% recall respectively.

Keywords: Cyber-threat intelligence (CTI), Denial of service (DoS), Intrusion Detection System (IDS), Intrusion Prevention System (IPS), Indicators of compromise (IoC), Network Intrusion Detection System (NIDS), Artificial Neural Network (ANN).

1. Introduction

Cyber-security is increasingly an essential component of today's modern world. As networks and systems are increasing very rapidly, protecting data from attacks is an important aspect of today's research. In recent times, protection from various cyber-attacks has become a challenging issue [1-7]. The current system, which comprises firewalls, data encryption techniques, and user authentication procedures, is insufficient to address the threats posed by modern sophisticated attackers. However, these security devices are unable to protect networks against cyber-attacks [8-10]. Artificial intelligence is playing an increasingly important role in this field, and it is now widely used in all industries [11-14].

Firewalls and Intrusion Detection/Prevention Systems (IDS/IPS) are examples of network security devices. The common occurrence of false positive alarms as well as failure to identify zero-day attacks, which destroy businesses are just a few of the issues that are known to plague existing IDS. Companies lose time in the investigative process due to the flaws in IDS backend engines. Deep packet inspection is conducted to detect malicious traffic in the network. Every packet that passes through it is examined and the payload is compared to signature databases. The request is blocked

if a match is discovered; otherwise, the network allows it to move on [15–18]. IDS are of two types as shown in Figure 1. A host intrusion system (HIDS) is installed on the host to identify attacks, while a network intrusion system (NIDS) is utilized for network-based activity. The NIDS come in two varieties. One of them is based on signatures. The second sort of detection is behavioral or anomaly-based. This kind is employed to identify unidentified attacks such as zero-day attacks [19–20]. Anomaly detection is the process of recognizing patterns in data that do not have predefined usual behavior [21].

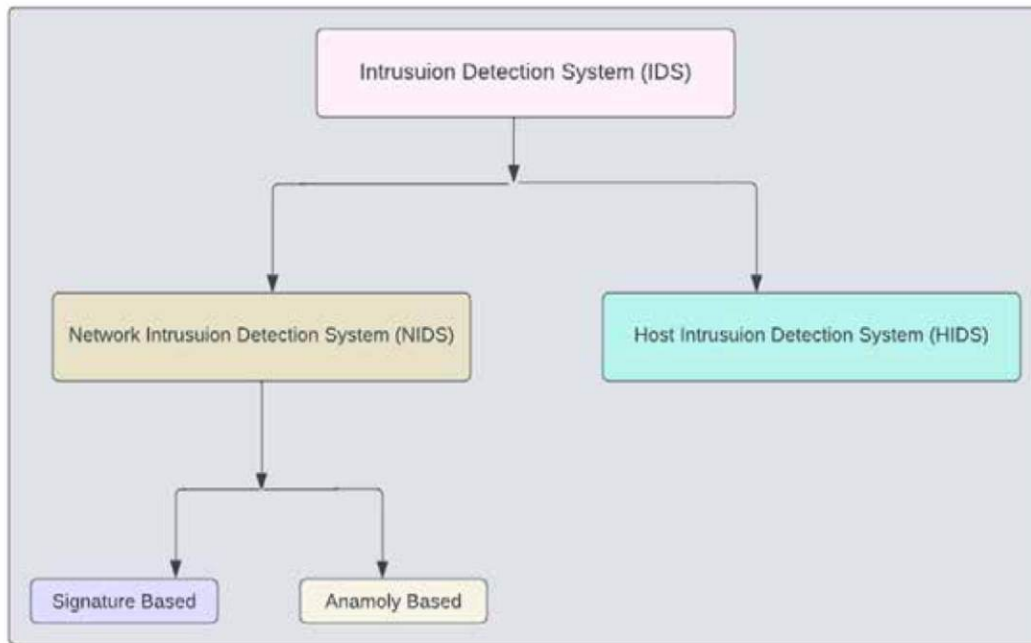


Figure 1. Types of IDS

To classify data into several categories modern-day machine learning techniques are used [22]. Normally traffic is classified into two types: normal and attack traffic. But according to some studies, there are five categories, attacks are further divided into four types: from remote to local, from user to root, probing assault, and DoS [23–26].

1.1 Role of Machine Learning in IDS

To protect against cyber-attacks, precise defense methods such as ML-based intrusion detection systems are required for better protection. They are being used as potential methods for detecting network attackers [27–30]. There is a need to categorize network attacks, despite major research efforts, IDS still struggles to improve detection accuracy [31–34]. ML/deep learning methods can be used in three ways: individually, hybrid, and ensemble-based. The performance of the machine learning technique was evaluated using several datasets. The most popular dataset for measuring performance is the NSL KDD Dataset [35–38].

1.2 Role of CTI in IDS

CTI provides multi-source databases that aid cyber defense mechanisms, allowing for comprehensive monitoring, identification, and response to online threats [39-41]. CTI feeds reveal how an attack occurred and who is behind it. These data feeds will be used to develop significant defensive security methods. The CTI feeds cover all major threat vectors, including websites, social media, bot IP addresses, malicious URLs, phishing URLs, spam, and harmful URLs. It enables organizations to decide how to respond to impending threats [42-44].

CTI generates threat feeds using both internal and external sources. Internal feeds include data from corporate security systems such as IDS/IPS, firewalls, and antivirus software. An external source could be a threat feed from a public (untrustworthy) source, such as an anti-malware domain, or a paid private source from several well-known and reputable security vendors.

Threat feeds are useful for organizations to protect against future attacks. Organizations are incorporating these threat feeds into their devices. Due to the sophisticated nature of the attacker, the attack's surface is continuously changing. There is a need to create an automated IDS mechanism that uses machine learning approaches to protect against assaults more accurately and precisely. CTI plays a vital role in providing updated threat feeds to security devices. [45-50].

This research investigation has made the following contributions:

- a. Research has concentrated on finding significant traits and obtaining useful information from datasets. Feature selection strategies are used which help to reduce dimensionality.
- b. The study work also contributes to normalizing the dataset. Some values in the dataset can affect the findings. So, normalization is performed on datasets.
- c. This investigation uses machine learning techniques for improved attack detection in IDS. This implementation has increased the performance metrics as compared to other techniques.

The following is the structure of the research paper. Part 2 includes a literature review, and section 3 describes the problem statement. Section 4 describes the datasets used for IDS analysis. The proposed methodology is outlined in Section 5. Part 6 contains the results, whereas Section 7 contains the conclusion and future work.

2. Literature Review

According to this study [51], a classifier strategy for NIDS employing the tree algorithm is used. The author proposes a combined tree classifier strategy for identifying network assaults. The author presents an IDS framework [52]. The author used a Bayesian classifier to find abnormalities in the network. The NSL-KDD dataset is used as a standard in this domain. According to this study [53], ML algorithms are utilized to detect security threats. This approach makes use of a support vector machine (SVM) to improve attack detection accuracy. The author presented a novel approach termed outlier detection to detect network intrusion [54]. The NSL-KDD dataset was used to validate the proposed approach.

This study [55] looked at the feasibility of combining fuzzy logic with machine learning approaches to detect intrusions. This research study [56] proposes an attack detection mechanism for IDS. When network flow exhibits anomalous behavior, this idea can help detect problems. The authors [57] introduced a novel paradigm for intrusion detection systems. The suggested study shows that using K-means clustering enhances IDS accuracy in identifying attacks. According to this study [58], entropy can detect anomalous network behavior, although at a high false rate. This study addresses the limitations of network entropy.

In this paper [59], the authors employed K-means and a naïve Bayes algorithm for IDS. When the K-means algorithm is used with naive Bayes, the detection rate increases while producing fewer false alarms. The authors conducted experiments with the Koyot 2006+ dataset. This research [60] provides a comprehensive review of anomaly-based detection, which uses single, hybrid, and ensemble machine learning models to assess distinct datasets. J48 and MLP classifier’s performance was evaluated for attack detection in IDS [61]. According to the results, J48 performed the best in detecting and categorizing all assaults in the NSL-KDD dataset. This study [62] carried out anomaly detection analysis and provided a comparative review of seven machine learning model performances on the Kyoto 2006+ dataset.

The authors presented [63] a hybrid system that employs two detection systems: abuse for signature or previously known forms of intrusions and anomaly for new and updated intrusions. Using the NSLKDD dataset, this study [64-65] assessed the performance of two supervised ML models, ANN and SVM. In this proposed study [66], a review is conducted for detecting attacks in IDS. According to this study [67], to detect intrusion threats in a computer network, four ML algorithms are evaluated on the KDD Cup dataset. Random forest and random tree algorithms performed the best on test datasets. This work looks at ML/DLNN models for intrusion detection systems [68–71]. Table 1 presents the outcomes of many previous methodologies.

Table 1. Results of Various Techniques

Author/ Year	Dataset	Technique	Results
A. Alzahrani et al. /2021	NSL-KDD	XGBoost	Precision 92% Recall 89% F1-Score 90%
V. Pai et al. /2021		Random Forest	Accuracy 91% Precision 92% Recall 90% F1-score 92%
A. Halimaa et al. /2019		Support Vector Machine	Accuracy 93%
K. Abu et al/2019	CSE-CICIDS-2018	ANN	Accuracy 91%
M. Fawa'reh et al. 2022		DNN-PCA	Accuracy 96%
J. Kim et. al. / 2019		CNN	Accuracy 95%
V. Kanimozhi et al. /2019		ANN, RF, KNN, SVM, Adaboost, NB	Accuracy 96%, Precision 90% Recall 95% F1- Score 90%
M. Amine et al. / 2019		DNN, RNN, CNN	Accuracy 93%

3. Problem Statement

With the intelligence and diversity of cyber threats increasing, traditional IDS are having problems detecting and mitigating attacks. The sheer volume and complexity of network traffic make it difficult for rule-based IDS to keep up with emerging threats. As a result, there is an urgent need to increase IDS capabilities by employing ML techniques for more precise threat detection. Because of the attack’s sophistication and increasing volume, it is difficult to detect them in real time. There

is a need to improve strategies for detecting attacks more precisely to make more accurate decisions concerning the detection of hostile activities.

4. Datasets for IDS Analysis

The KDD Cup 99 dataset was created at the fifth international conference on knowledge discovery and data mining. It was developed at the Network Security Laboratory-KDD (NSL-KDD). It contains forty-one features [22]. The data includes records from KDDTrain+, KDDTest21+, and KDD Test+, totaling 125,973, 11,850, and 22,544. The Aegean Wi-Fi Intrusion Dataset (AWID) is the most widely used and publicly available IDS dataset. AWID is detected by character data as well as an imbalance between attack and regular data.

The Yahoo Web Scope S5 includes labeled anomalous events from both genuine and bogus time series. It tests how well various anomaly types, such as outliers and change points can be identified. The Numenta Anomaly Benchmark (NAB) dataset evaluates approaches for detecting anomalies in streaming web applications. It includes more than 50 annotated real-world and synthetic time series data files. The Kyoto 2006+ dataset is based on actual network traffic data collected over three years and classified as normal or attack traffic.

The UNSW-NB 15 dataset was generated by the Australian Centre for Cyber Security (ACCS) and mixes genuine current normal activities with synthetic contemporary attack behaviors. The UNSW Canberra Cyber Range Lab generated the Bot-IoT dataset by simulating a network. The traffic consists of both standard and botnet traffic. The ISCX IDS 2012 dataset was produced in 2012. The essential notion is built on profiles, lower-level network parts, and precise intrusion descriptions. The CSE-CIC-IDS2018 dataset pioneered the concept of a profile. It has amassed 16,000,000 occurrences in ten days. This is the most recent public big data intrusion detection dataset, and it includes a wide spectrum of attack strategies.

5. Proposed Methodology

The proposed methodology compares two datasets: NSL KDD and CSE-CIC-IDS2018. These are the two most utilized datasets in IDS analysis to detect attacks.

5.1 Methodology for NSL-KDD Dataset

The suggested approach for evaluating the NSL-KDD dataset is divided into three main parts. In the first stage, data transformation techniques are used. The second phase entails decreasing features. The third phase uses classification techniques like support vector machines, random forest, and decision trees to detect risks.

Figure 2 displays the methodology for the NSL-KDD dataset. Three phases make up the proposed methodology. The data preprocessing phase is the first stage. Using data transformation techniques like label encoder, the data set is transformed into numerical values at this phase. Data transformation techniques are used to convert the data to a single numerical value since machine learning algorithms perform best on single-value datasets. Feature reduction is the second stage. During this phase, feature reduction techniques like PCA are used to minimize the feature set. Forty-one features are reduced to fourteen in this phase. Computational power grows when more features are used in the dataset. Hence, feature reduction techniques are utilized to save computational resources. The third phase involves using ML methods to classify data. This step uses a decision tree, random forest, and SVM algorithms to classify data. The training and testing

data sets are split 80:20. Machine learning techniques classify data as assaults or normal traffic.

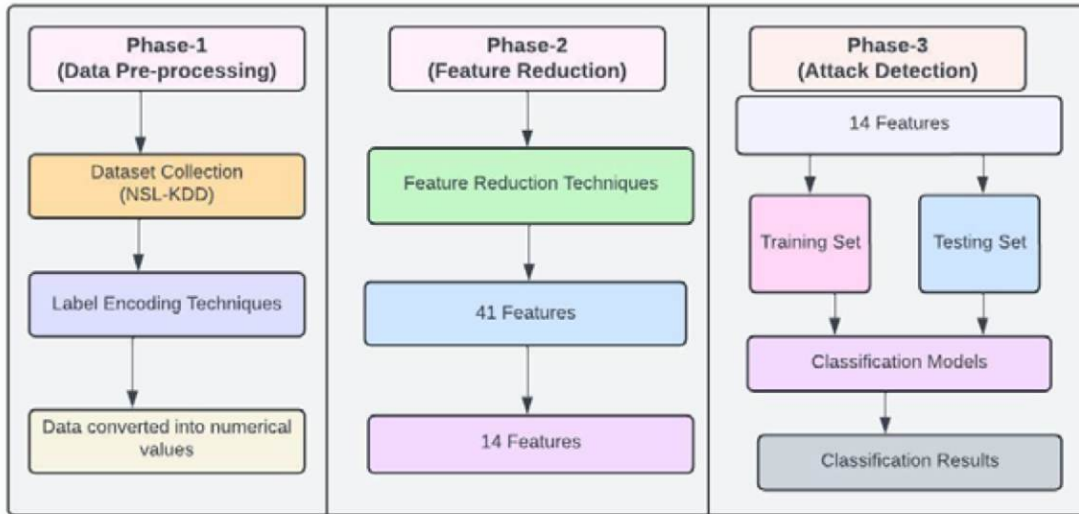


Figure2. Proposed Methodology (NSL-KDD Dataset)

Phase-1

This dataset contains numerical and nominal values. During this phase, all values are converted into numerical representation. This transformation is carried out utilizing a label encoder. It is employed because it is the most often used method. Converting values to a single value has the advantage of delivering correct results, as machine learning algorithms work best with specific types of values.

Phase-2

The subsequent stage is the feature reduction phase. The purpose of feature reduction is to reduce the dataset's forty-one features, which require greater processing resources to compute their values. The literature employs a range of feature reduction techniques, including genetic algorithms, linear discriminant analysis (LDA), principal component analysis (PCA), information gain, and generalized discriminant analysis. PCA is the most popular feature reduction method in the world today. PCA is used here because it is simple to calculate and yields consistent results. Computing systems find it simple to solve problems. Reduced dimensionality enhances the performance of machine learning algorithms. One benefit of using PCA is that it reduces data noise.

Approaches such as the genetic algorithm are computationally complex. Data with many different dimensions is difficult to represent; so, PCA makes visualization of data easier by reducing the dimensions. The proposed study's feature set consists of 41 features. PCA compresses the original forty-one feature set to fourteen main features. A threshold is established, and values more than 0.60 are classified as a feature. In this regard, fourteen feature sets have been selected. By reducing the number of data set features, feature reduction algorithms improve system performance and require less processing power. Table 2 shows the best 14-feature set recovered by the PCA.

Table 2. Optimal Feature Set

Sr.#	Feature	Sr.#	Feature
1.	Protocol_type	8.	Srv_count
2.	Service	9.	Duration
3.	Src_bytes	10.	Dst_host_count
4.	Dst_bytes	11.	Wrong_fragment
5.	Num_failed_logins	12.	Dst_host_srv_count
6.	Root_shell	13.	urgent
7.	Count	14.	Logged_in

Phase-3

The following stage is to apply classification algorithms on the data extracted from phase 2 with fourteen features. Classification techniques include SVM, RF, and DT.

Figure 3 depicts a flow diagram. The system accepts the NSL-KDD data set as input. Data transformation techniques are used to convert data into a single numerical value. The features in the data set are then reduced using feature reduction techniques. Following feature reduction methods, classification algorithms are used to distinguish between legitimate and malicious traffic.

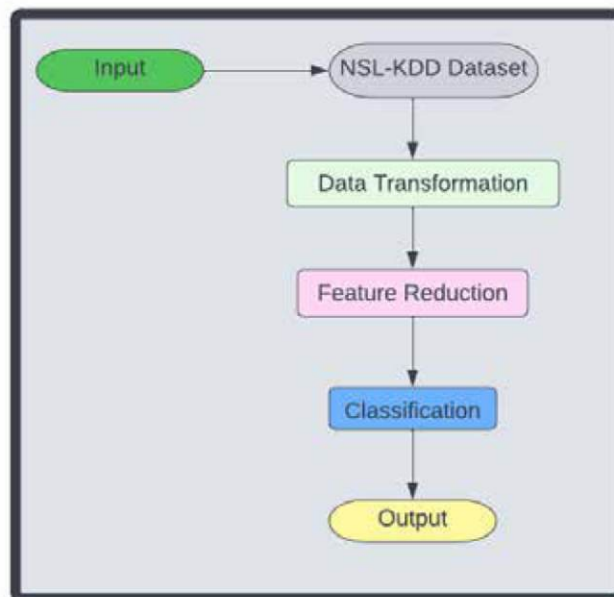


Figure 3. Flow Diagram

5.2 Proposed Methodology for CSE-CIC-IDS2018 Dataset

The proposed technique for analyzing the CSE-CIC-IDS2018 dataset is separated into three stages. The first stage is normalization, which employs methods such as z-score and min-max normalization. The second phase involves feature reduction techniques such as PCA, whereas the third employs classification methods such as SVM, RF, and DT.

Figure 4 depicts the optimal approach for the CIC-IDS2018 dataset. The proposed methodology is made up of three steps. The normalization phase is the first phase. This phase entails normalizing the dataset with techniques like z-score. Normalization is a common method for getting data ready

for machine learning. Normalization is the process of converting numeric column values in a dataset to a standard scale while retaining information and preventing value range distortion. The second step is all about reduction. The feature set is reduced at this step using feature reduction techniques such as PCA.

This phase reduced the number of features from 81 to 53. The processing capability of a data set grows as more features are added. Thus, feature reduction techniques are employed to save computational resources. In the third phase, data is classified using machine learning approaches. In this step, data is classified using decision trees, random forests, and SVM algorithms. The training and testing data sets are split 80:20. Machine learning techniques classify the data as either an attack or normal traffic.

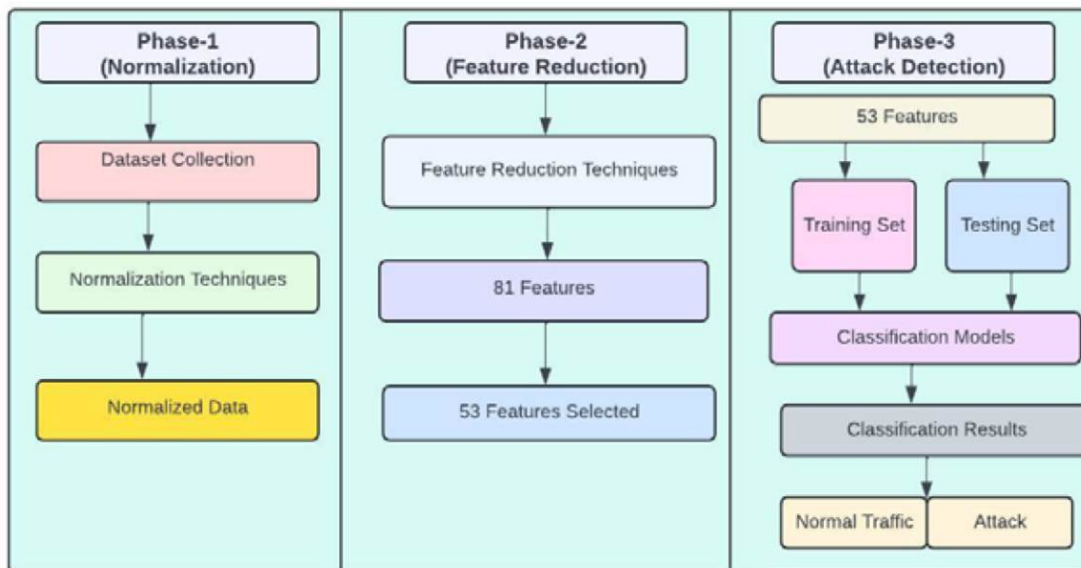


Figure 4. Proposed Methodology (CSE-CIC-IDS2018 Dataset)

Phase-1

The first step is to standardize the data. Because the values in certain columns are high. Normalization procedures are employed to balance the values in the data. The advantage of utilizing normalization procedures is that they equalize all the column values. For this reason, the z-score is used.

Phase-2

In the second phase normalized dataset is used for feature reduction as it consists of eighty-one features that require more computational power and resources for utilization. For feature reduction, a technique like PCA is used. The threshold is set at 0.60. Values higher than this threshold are selected. Eighty feature sets are reduced to fifty-three feature sets.

Phase-3

The following stage is to apply a classification algorithm to the phase 2 data, which contains fourteen features. SVM, RF, and DT are the classification algorithms employed.

Figure 5 depicts a flow diagram. The CSE-CIC-IDS2018 dataset is utilized as the system input. Normalization processes are used to make the data more consistent. The dataset features are then reduced using feature reduction techniques. Following feature reduction methods, classification

algorithms are used to differentiate between legitimate and malicious traffic.

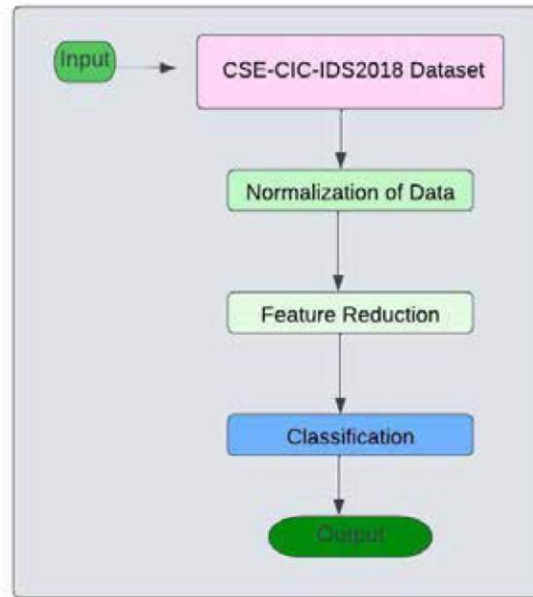


Figure 5. Flow Diagram

6. Results

Several performance evaluation criteria, including Recall, Accuracy, and Precision, are used for experimentation. Accuracy measures a model's overall performance. Relying primarily on accuracy is not an original concept. Precision determines the classifier's expected positive results among all positive discoveries. Sensitivity is sometimes referred to as Recall. Precision and Recall are better employed together than separately because they are ineffective performance measures when used alone. The confusion matrix for the NSL KDD dataset is depicted in Figure 6. This matrix shows both the expected and actual values. The model's anticipated true values compared to predicted false values.

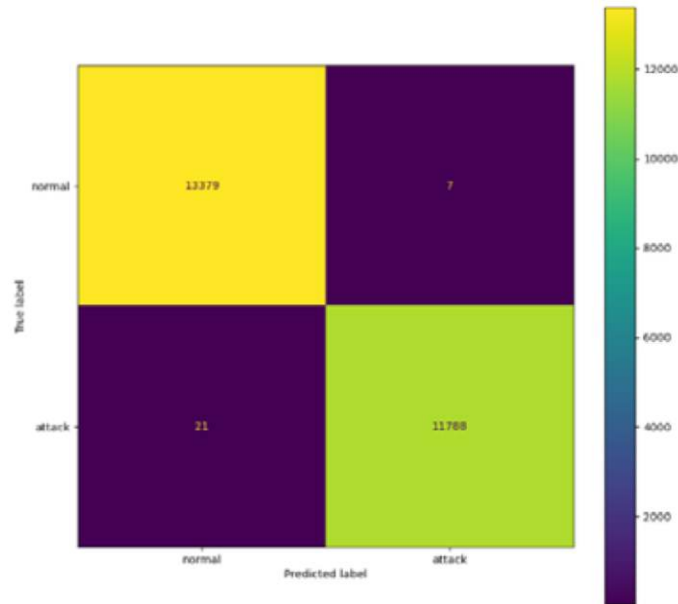


Figure 6. Confusion Matrix (NSL-KDD Dataset)

Using the NSL-KDD dataset, the proposed algorithm achieves 95% accuracy, which outperforms earlier methods. Random forest yields 96% accuracy, 94% precision, and 94% recall. SVM delivers 94% accuracy, 92% precision, and 92% recall rates. The decision tree achieves 92% accuracy, 92% precision, and 91% recall.

The experiment employs cross-validation with a value of k=10. The training and testing datasets have an 80:20 ratio. Using the CSE-CIC-IDS2018 dataset, the proposed methodology surpasses earlier methods, with an accuracy of 98%. Using random forest, we get 98% accuracy, 97% precision, and 96% recall. SVM yields 94% accuracy, 95% precision, and 95% recall, respectively. The decision tree has 93% accuracy, 94% precision, and 94% recall, respectively. The implementation language is Python. Anaconda is used to create an integrated development environment. The implementation testbed is powered by a Core-I-7 CPU with 16 GB of RAM. Figures 7 and 8 compare the results with the datasets NSL-KDD and CSE-CIC-IDS2018.

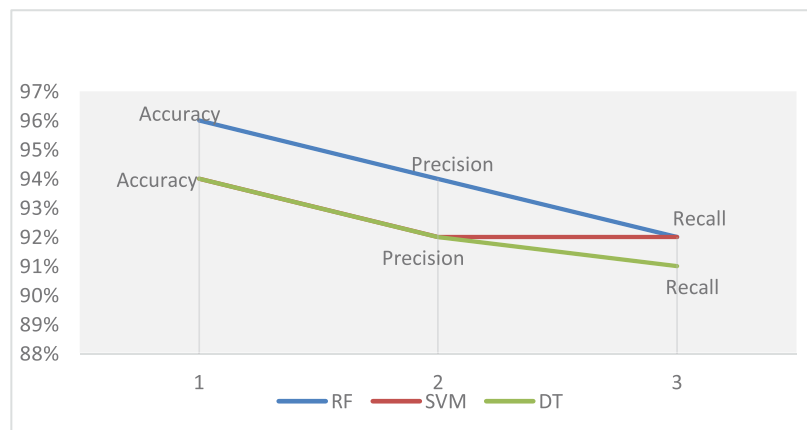


Figure 7. Results (NSL-KDD Dataset)

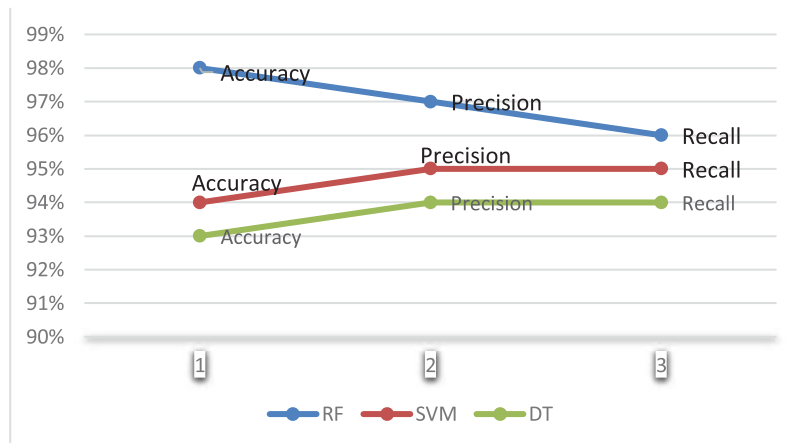


Figure 8. Results (CSE-CIC-IDS2018 Dataset)

7. Conclusion and Future Work

The rate of cybercrime is rapidly increasing, posing a significant disadvantage to technology. There are many attacks and methods by which attackers can breach systems. To secure systems from such attackers, researchers created several solutions based on machine learning algorithms, which are crucial for detecting and safeguarding assets from a variety of threats. Using ML methodologies, this paper proposed a strategy for more precisely detecting attacks in IDS. The suggested approach employs two of the most extensively used datasets for experimentation. This methodology has an overall accuracy of 96% for the NSL-KDD and 98% for the CSE-CIC-IDS2018 dataset. The suggested system identifies network attacks with greater accuracy and precision than previous methods. Deep learning techniques will be utilized in the future to improve categorization accuracy.

References

- Conklin, Art and White, Gregory B, "E-government and cyber security: the role of cyber security exercises", Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), IEEE, vol4, pp79b-79b, the Year 2006.
- Leuprecht, Christian and Skillicorn, David B and Tait, Victoria E, "Beyond the Castle Model of cyber-risk and cyber-security", Government Information Quarterly, volume 33, pp 250-257, the year 2016.
- Zwilling, Moti and Klien, Galit and Lesjak, Duan and Wiechetek, and Cetin, Fatih and Basim, Hamdullah Nejat, "Cyber security awareness, knowledge and behavior: A comparative study", Journal of Computer Information Systems, volume 62, pp 82-97, the year 2022.
- Rajasekharaiah, KM and Dule, Chhaya S and Sudarshan, E, "Cyber security challenges and its emerging trends on latest technologies", IOP Conference Series: Materials Science and Engineering, volume 981, pp 022062, the year 2020.
- Tonge, Atul M and Kasture, Suraj S and Chaudhari, Surbhi R, "Cyber security: challenges for society-literature review", IOSR Journal of Computer Engineering, volume 2, pp 67-75, 2013.
- Von Solms, Rossouw and Van Niekerk, Johan, "From information security to cyber security", computers & security, volume 38, pages 97-102, the year 2013.
- McNeese, Michael and Cooke, Nancy J and D'Amico, Anita and Endsley, Mica R and Gonzalez, Cleotilde and Roth, Emilie and Salas, Eduardo, "Perspectives on the role of cognition in cyber security", Proceedings of the Human Factors and Ergonomics Society Annual Meeting, volume 56, pages 268-271, the year 2012.

8. Choo, Kim-Kwang Raymond, "The cyber threat landscape: Challenges and future research directions", *Computers & Security*, volume 30, pp719-731, the year 2011.
9. Spence, Aaron and Bangay, Shaun, "Security beyond cybersecurity: side-channel attacks against non-cyber systems and their countermeasures", *International Journal of Information Security*, volume= 21, pp 437-453, 2022.
10. Achar, Sandesh," Cloud Computing Security for Multi-Cloud Service Providers: Controls and Techniques in our Modern Threat Landscape", *International Journal of Computer and Systems Engineering*, volume=16, pages 379-384,2022.
11. Rowe, Dale C. and Lunt, Barry M., and Ekstrom, Joseph J, "The role of cyber-security in information technology education", *Proceedings of the 2011 conference on Information technology education*, pp 113-122, 2011.
12. Ukwandu, Elochukwu and Ben-Farah, Mohamed Amine and Hindy, Hanan, and Bures, Miroslav and Atkinson, Robert and Tachtatzis, Christos and Andonovic, Ivan and Bellekens, Xavier, *Cyber-security challenges in the aviation industry: A review of current and future trends*, *Information, MDPI*, volume 13, pp 146, 2022.
13. Mahmood, Samreen and Chadhar, Mehmood and Firmin, Selena, "Cybersecurity challenges in blockchain technology: A scoping review", *Human Behavior and Emerging Technologies*, Hindawi, volume 2022, 2022.
14. Akpan, Frank and Bendiab, Gueltoum and Shiaeles, Stavros and Karamperidis, Stavros and Michaloliakos, Michalis, "Cybersecurity challenges in the maritime sector" *Network*, MDPI volume2, pp 123-138, 2022.
15. Denning, Dorothy E, "An intrusion-detection model", *IEEE Transactions on Software Engineering*, pp 222-232, 1987.
16. Roschke, Sebastian and Cheng, Feng and Meinel, Christoph, "Intrusion detection in the cloud", 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, IEEE, pp729-734,2009.
17. Effendy, David Ahmad and Kusriani, Kusriani, and Sudarmawan, Sudarmawan, "Classification of the intrusion detection system (IDS) based on the computer network. 2017 2nd International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), IEEE, pp 90-94, 2017.
18. Uppal, Hussain Ahmad Madni and Javed, Memoona and Arshad, M, "An overview of the intrusion detection system (IDS) along with its commonly used techniques and classifications", *International Journal of Computer Science and Telecommunications*, Citeseer, volume 5, pp 20-24, 2014.
19. Ashoor, Asmaa Shaker and Gore, Sharad, "Importance of intrusion detection system (IDS)", *International Journal of Scientific and Engineering Research*, volume 2, pp 1-4,2011.
20. Liao, Hung-Jen and Lin, Chun-Hung Richard and Lin, Ying-Chih and Tung, Kuang-Yuan, "Intrusion detection system: A comprehensive review", *Journal of Network and Computer Applications*, volume 36, pp 16-24, 2013.
21. Wu, Yu-Sung and Foo, Bingrui and Mei, Yongguo and Bagchi, Saurabh, "Collaborative intrusion detection system (CIDS): a framework for accurate and efficient IDS", 19th Annual Computer Security Applications Conference, 2003. *Proceedings*, IEEE, pp 234-244, 2003.
22. Khraisat, Ansam and Gondal, Iqbal and Vamplew, Peter and Kamruzzaman, Joarder, "Survey of intrusion detection systems: techniques, datasets, and challenges", *Cybersecurity*, Springer, volume 2, pp 1-22,2019.
23. Kr. gel, Christopher and Toth, Thomas and Kirda, Engin, "Service-specific anomaly detection for network intrusion detection", *Proceedings of the 2002 ACM symposium on Applied computing*, pp 201-208, 2002.
24. Hnamte, Vanlalruata and Hussain, Jamal, "An Extensive Survey on Intrusion Detection Systems: Datasets and Challenges for Modern Scenario", 2021 3rd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), IEEE, pp 1-10, 2021.
25. Umer, Muhammad Fahad, and Sher, Muhammad, and Bi, Yaxin, "Flow-based intrusion detection: Techniques and challenges", *Computers & Security*, volume70, pp 238-254,2017.
26. Hindy, Hanan and Brosset, David and Bayne, Ethan and Seem, Amar and Tachtatzis, Christos and Atkinson, Robert and Bellekens, Xavier, "A taxonomy and survey of intrusion detection system design techniques, network threats and datasets", 2018.
27. Azizjon, Meliboev and Jumabek, Alikhanov and Kim, Wooseong, "2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)}, IEEE, pp 218-224,2020.
28. Panigrahi, Ranjit and Borah, Samarjeet and Bhoi, Akash Kumar and Ijaz, Muhammad Fazal and Pramanik, Moumita and Kumar, Yogesh and Jhaveri, Rutvij H, "Mathematics, MDPI, volume 9, pp 751, 2021.
29. Balyan, Amit Kumar and Ahuja, Sachin and Lilhore, Umesh Kumar and Sharma, Sanjeev Kumar and Manoharan, Poongodi, and Algarni, Abeer D and Elmannai, Hela and Raahemifar, Kaamran, "A hybrid intrusion detection model using ega-pso and improved random forest method", *Sensors*, MDPI, volume 22, pp 5986, 2022.

30. Ashraf, Javed and Moustafa, Nour and Khurshid, Hasnat and Debie, Essam and Haider, Waqas and Wahab, Abdul, "A review of intrusion detection systems using machine and deep learning in the internet of things: Challenges, solutions, and future directions", *Electronics*, MDPI, volume 9, pp 1177, 2020.
31. Kasongo, Sydney Mambwe and Sun, Yanxia, "A deep learning method with filter-based feature engineering for wireless intrusion detection system", *IEEE Access*, volume 7, pp 38597-38607, 2019.
32. Salem, Maher and Al-Tamimi, Abdel-Karim, "A Novel Threat Intelligence Detection Model Using Neural Networks", *IEEE Access*, volume 10, pp 131229-131245, 2022.
33. RM, SwarnaPriya and Maddikunta, Praveen Kumar Reddy and Parimala, M and Koppu, Srinivas and Gadepalli, Thippa Reddy and Chowdhary, Chiranjilal, and Alazab, Mamoun, "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture, *Computer Communications*, Volume 160, pp 139-149, 2020.
34. Kumar, Vikash and Sinha, Ditipriya and Das, Ayan Kumar and Pandey, Subhash Chandra and Goswami, RadhaTamal, "An integrated rule-based intrusion detection system: analysis on UNSW-NB15 data set and the real-time online dataset", *Cluster Computing*, Springer, volume 23, pp 1397-1418, 2020.
35. Alohal, Manal Abdullah and Al-Wesabi, Fahd N and Hilal, Anwer Mustafa and Goel, Shalini, and Gupta, Deepak and Khanna, Ashish, "Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment", *Cognitive Neurodynamic*, Springer, volume 16, pp 1045-1057, 2022.
36. Guarascio, Massimo and Cassavia, Nunziato and Pisani, Francesco Sergio and Manco, Giuseppe, "Boosting cyber-threat intelligence via collaborative intrusion detection", *Future Generation Computer Systems*, volume 135, pp 30-43, 2022.
37. Li, XuKui and Chen, Wei and Zhang, Qianru and Wu, Lifa, "Building auto-encoder intrusion detection system based on random forest feature selection, *Computers & Security*, volume 95, pp 101851, 2020.
38. Asif, Muhammad and Abbas, Sagheer and Khan, MA and Fatima, Areej and Khan, Muhammad Adnan and Lee, Sang-Woong, "MapReduce based intelligent model for intrusion detection using machine learning technique", *Journal of King Saud University-Computer and Information Sciences*, 2021.
39. T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, "Cyber threat intelligence sharing: Survey and research directions," *Computers & Security*, vol. 87, p. 101589, 2019.
40. T. D. Wagner, E. Palomar, K. Mahbub, and A. E. Abdallah, "A novel trust taxonomy for shared cyber threat intelligence," *Security and Communication Networks*, vol. 2018, 2018.
41. V. Mavroeidis and S. Bromander, "Cyber threat intelligence model: an evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence," in *2017 European Intelligence and Security Informatics Conference (EISIC)*. IEEE, 2017, pp. 91-98.
42. M. Conti, T. Dargahi, and A. Dehghantaha, "Cyber threat intelligence: challenges and opportunities," in *Cyber Threat Intelligence*. Springer, 2018, pp. 1-6.
43. Gartner, "2021 Gartner," <https://www.gartner.com>, 2021.
44. R. Brown and R. M. Lee, "The evolution of cyber threat intelligence (cti)": 2019 sans cti survey," *SANS Institute: Singapore*, 2019.
45. Tounsi, Wiem and Rais, Helmi, "A survey on technical threat intelligence in the age of sophisticated cyber-attacks", *Computers & Security*, volume 72, pp 212-233, 2018.
46. Ramsdale, Andrew and Shiaeles, Stavros and Kolokotronis, Nicholas, "A comparative analysis of cyber-threat intelligence sources, formats, and languages", *Electronics*, volume 9, pp 824, 2020.
47. Berndt, Anzel and Ophoff, Jacques, "Exploring the value of a cyber threat intelligence function in an organization", *Information Security Education. Information Security in Action: 13th IFIP WG 11.8 World Conference, WISE 13, Maribor, Slovenia, September 21--23, 2020, Proceedings 13*, Springer, pp 96-109, 2020.
48. Zibak, Adam and Simpson, Andrew, "Cyber threat information sharing: Perceived benefits and barriers", *Proceedings of the 14th International Conference on Availability, Reliability, and Security*, pp 1-9 2019.
49. Samtani, Sagar and Abate, Maggie and Benjamin, Victor and Li, Weifeng, "Cybersecurity as an industry: A cyber threat intelligence perspective", *The Palgrave Handbook of International Cybercrime and Cyberdeviance*, Springer, pp 135-154, 2020.
50. Zibak, Adam and Sauerwein, Clemens and Simpson, Andrew, "A success model for cyber threat intelligence management platforms", *Computers & Security*, volume 111, pp 102466, 2021.
51. Kevric, J., Jukic, S. Subasi, A. An effective combining classifier approach using tree algorithms for network intrusion detection. *Neural Computing Applications* 28, 1051–1058 (2017).
52. Kabir, MdReazul, Abdur Rahman Onik, and TanvirSamad. "A network intrusion detection framework based on Bayesian network using wrapper approach." *International Journal of Computer Applications* 166.4 (2017).

53. Hagos, DestaHaileselassie, et al." Enhancing security attacks analysis using regularized machine learning techniques." 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA). IEEE, 2017
54. DivyaGoyal, Research Scholar Hardeep Singh, A.P. Dept. CSE at LPU, Jalandhar. Paper on Machine learning Techniques: Outlier Detection and Text summarization, International Journal of Scientific Engineering Research, Volume 5, Issue 3, March 2014 223
55. IJCSNS International Journal: Intrusion Detection Using Machine Learning along Fuzzy Logic and Genetic Algorithms, Y. Dhanalakshmi and Dr. Ramesh Babu, Dept of Computer Science Engineering Acharya Nagarjuna University, Guntur, A.P. India.
56. Chitrakar, Roshan, and Chuanhe Huang." Anomaly-based intrusion detection using hybrid learning approach of combining k-medoids clustering and naive Bayes classification." 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing. IEEE, 2012
57. Duque, Solane, and MohdNizam bin Omar." Using data mining algorithms for developing a model for intrusion detection system (IDS)." Procedia Computer Science 61 (2015): 46-51.
58. Agarwal, Basant, and Namita Mittal." Hybrid approach for detection of anomaly network traffic using data mining techniques." Procedia Technology 6 (2012): 996-1003
59. Muda, Z. Mohamed, WarusiaSulaiman, md nasirUdzir, Nur. (2016). K-Means Clustering and Naive Bayes Classification for Intrusion Detection. Journal of IT in Asia. 4. 13-25. 10.33736/jita.45.2014.
60. U. S. Musa, M. Chhabra, A. Ali and M. Kaur," Intrusion Detection System using Machine Learning Techniques: A Review," 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 149-155, doi: 10.1109/ICOSEC49089.2020.9215333.
61. Alkasassbeh and Almseidin. (2018). Machine Learning Methods for Network Intrusions. International Conference on Computing, Communication (ICCCNT). Arxiv.
62. Marzia Z. and Chung-Horng L. (2018). Evaluation of Machine Learning Techniques for Network Intrusion Detection. IEEE. (pp. 1-5)
63. Dutt t I. et al. (2018). Real-Time Hybrid Intrusion Detection System. International Conference on Communication, Devices and Networking (ICCDN). (pp. 885-894). Springer.
64. Kazi A., Billal M. and Mahbubur R. (2019). Network Intrusion Detection using Supervised Machine Learning Technique with feature selection. International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST). (pp. 643-646). IEEE.
65. Rajagopal S., Poornima P. K. and Kat iganere S. H. (2020). A Stacking Ensemble for Network Intrusion Detection using Heterogeneous Datasets. Journal of Security and Communication Networks. Hindawi.
66. S. Thapa and A.D Mailewa (2020). The Role of Intrusion Detection/Prevention Systems in Modern Computer Networks: A Review. Conference: Midwest Instruction and Computing Symposium (MICS). Wisconsin, USA. Volume: fifty-three. (pp. 1-14).
67. Chibuzor John Ugochukwu, E. O Bennett. An Intrusion Detection System Using Machine Learning Algorithm Department of Computer Science, International Journal of Computer Science and Mathematical Theory ISSN 2545-5699 Vol. 4 No.1 2018.
68. Alqahtani H., Sarker I.H., Kalim A., Minhaz Hossain S.M., Ikhlq S., Hossain S. (2020) Cyber Intrusion Detection Using Machine Learning Classification Techniques. In: Chaubey N., Parikh S., Amin K. (eds) Computing Science, Communication and Security. COMS2 2020. Communications in Computer and Information Science, vol 1235. Springer, Singapore. https://doi.org/10.1007/978-981-15-6648-6_10.
69. Xin, Y., et al.: Machine learning and deep learning methods for cybersecurity. IEEE Access 6, 35365–35381 (2018).
70. Ferrag, Maglaras, Moschoyiannis, Janicke (2019). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study, Journal of Information Security and Applications.
71. Singh, Geeta and Khare, Neelu, A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques, International Journal of Computers and Applications, 2021.

Predicting Student Performance Using Educational Data Mining: A Review

Veena Kumari^a, Areej Fatemah Meghji^{a*}, Rohma Qadir^a, Urooj Oad^a

^aDepartment of Software Engineering, Mehran University of Engineering and Technology, Pakistan

vina.mehrani@gmail.com, areej.fatemah@faculty.muett.edu.pk, zainabqadir97@gmail.com,
uroojgianchand@gmail.com

Farhan Bashir Shaikh^b

^bFaculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Malaysia

farhanb@utar.edu.my

*Corresponding Author: Areej Fatemah Meghji areej.fatemah@faculty.muett.edu.pk

Abstract

Educational Data Mining (EDM) strategies facilitate the efficient and in-depth analysis of student data. EDM provides useful insights into comprehending student learning patterns and identifying factors that influence academic success. This review aims to evaluate the efficacy of classification algorithms popularly explored in EDM for predicting student performance and identifying common trends in existing EDM research. The review follows a systematic approach, relevant research articles have been cited following an inclusion and exclusion criteria to ensure the selection of studies that specifically address the use of EDM techniques for predicting student academic achievement. According to the review findings, most researchers have utilized the features of cumulative grade point average, internal and external assessment, and demographic information to predict student performance. The most common techniques in EDM for predicting students' performance are Naïve Bayes and Decision Trees. The review also focuses on the potential for bias, key examination of challenges, and possible future directions in the field. In the context of student performance prediction, ethical considerations regarding privacy, data handling, and the interpretation of results are also identified.

Keywords: classification, educational data mining, decision tree, academic achievement.

1. Introduction

Educational institutions have always collected vast amounts of student data [1]. These institutes are constantly trying to find the most effective ways to store, process, and make use of this data to better understand how students learn [2]. Getting insights from this data can greatly aid educational institutes in shaping pedagogical policies and devise strategies for fostering a student-centric environment of learning [3]. Educational data mining (EDM) is an emerging field of research tasked with analyzing, developing, and employing computational techniques to

uncover hidden patterns in large learning or educational data sets, which can be difficult to evaluate due to their size. This developing branch of knowledge and data mining explores real-time educational databases ranging from student admissions, registration, and examination, to course management systems like Moodle and Blackboard. The main goal of research using EDM is to analyze student academic achievement at different levels of their educational journey such as school, college, and university [2]. One key area of research within EDM is predicting student performance at an early stage during their education [3].

Classification is a popular, supervised data mining method that relies on learning from historic, labeled data; it finds patterns in the data to generate a model. The generated model is then used to make predictions for new instances of data by classifying them into pre-defined labels or classes based on the discovered patterns within the data [1]. Decision trees, Bayesian networks, neural networks, and K-nearest neighbor (KNN) are some of the often-employed approaches of classification [3].

Classification has been used to target the prediction of several facets of education including the prediction of courses that have a significant impact on final degree level performance [1], or factors that cause a fluctuation in the grade point average of a student [4]. An interesting area of research is the analysis of student data to categorize students into classes of learners [2]. These techniques concentrate on analyzing student educational data, which represents their academic performance, and developing clear rules to aid students in their future academic performance [5]. Universities have been analyzing educational data such as enrollment data, student academic data, student learning data, feedback data, and many other forms of data, using varied classification approaches to provide a university with the necessary knowledge to more effectively plan for student enrolment, avoid student dropouts, detection of students at risk of failure, and resource allocation with a precise approximation [6], [7]. The outcome of EDM is intended to help pedagogical decision making. An insight into the factors affecting student performance can help educational institutes mitigate those factors to boost student performance, which will directly boost the overall performance of an educational institution. This review presents a picture of the use of EDM towards the prediction of student academic performance. The goals of the systematic review are to:

1. Ascertain the features or attributes that are important for analyzing a student's performance.
2. Discover the most preferred EDM methods for predicting student performance.
3. Identify the challenges facing EDM research.

The review is structured as follows: Section 2 focuses on the research methodology including the research questions to be addressed, a search strategy to extract relevant previous studies, and a review conducting process. Section 3 provides the proposed research questions in parallel with the results analysis of the review, followed by a discussion in Section 4. The conclusion of the review is presented in Section 5.

2. Research Methodology

Figure 1 represents the research methodology followed to conduct this literature review.

1. The plan phase of the review included clearly identifying the scope or boundary of the review,

outlining the research questions that the review would address, and creating a review protocol that would outline the fundamental review procedure. This step also included establishing the sources for the collection of the studies, setting a criterion for inclusion/exclusion, and defining the process through which this criterion would be applied.

2. The second phase comprised the actual conduct of the review. This included extracting information based on the research questions defined in step-1; the gathered information was also assessed and tabulated.
3. The reporting phase involved presenting the findings of the review. The analysis was written up during this phase, ensuring the review's credibility and usefulness.

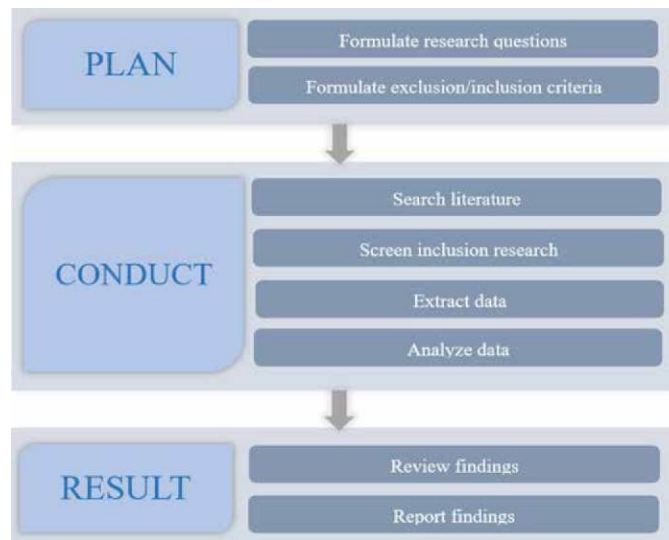


Figure1. Research Methodology

2.1 Research Questions

The right research questions are crucial to understanding the current study on performance prediction for students. The criteria for the study have been outlined in Table 1 following the Kitchenham et al. approach of identifying the Population, Intervention, Outcome, and Context [8].

Table 1. Research question criteria

Criterion	Description
Population	University students
Context	Empirical studies including preliminary studies, case studies, comparative study
Intervention	EDM methods for student performance prediction
Outcome	Classifier accuracy, most used classification algorithms, parameters used for performance prediction

Three research questions were proposed for this review:

RQ1. What are the most essential factors explored when forecasting student performance?

RQ2. Which classification techniques or algorithms are commonly used in EDM research with the intent of predicting student performance?

RQ3. What are the limitations or challenges in using EDM for predicting student performance?

2.2 Search Strategy

It is important to have a well-thought-out search strategy for a systematic literature review. This ensures finding the most relevant work from the massive pool of research. A thorough search was made for research publications that address the suggested research questions. The search was carried out in four knowledge sources namely IEEE Xplore, ACM Digital Library, Science Direct, and Google Scholar to extract relevant research papers. The text strings used to carryout the research are mentioned in Table 2.

Table 2. Search strings executed on knowledge sources

Knowledge Source	Search Strings
IEEE Xplore, ACM Digital Library, ScienceDirect, Google Scholar	“Student performance”, “Educational data mining methods”, “Student performance prediction”, “Educational data mining techniques”, “EDM student performance”, “EDM classification”

Items extracted during the search include preliminary empirical studies such as journal articles, case studies, comparative studies, workshop papers, and conference papers. Based on the search strings outlined in Table 2, a total of 124 articles were shortlisted initially as depicted in Table 3.

Table 3. Quantity of papers acquired from knowledge sources

Knowledge Source	Number of Research Articles
IEEE	28
ACM	07
Science Direct	16
Google Scholar	73

After formulating the research questions, selecting knowledge bases, and separating the research papers based on the search strings, an important step was to execute the inclusion/exclusion criterion. The formulation of a research selection process is an important step of the review as it helps set the scope and boundary of the review. It also helps guide the review process by allowing the researcher to consider or disregard a paper based on a set of pre-established criterion. The inclusion and exclusion criterion for the conducted review have been outlined in

Table 4.

Table 4. Exclusion and inclusion criteria

S.no	Inclusion Criteria	Exclusion Criteria
1	Nature of the research should be empirical	Non-empirical studies
2	Papers must be fully accessible	Papers that are not accessible
3	Papers must be published between 2014 to 2024	Papers published before 2014
4	The paper must include EDM techniques and findings	Paper that addresses EDM techniques in general
5	Papers that address the research questions	Papers not addressing the research questions

Figure 2 outlines the detailed paper selection process and the articles dropped at each stage during the conduct of this review. To ensure the quality of the research, the article inclusion criteria have been firmly followed. After the initial selection, the studies were foremost screened based on the title of the research. After the first title screening, 42 papers were eliminated. The abstracts of the selected papers were then scrutinized. All the papers were thoroughly examined to exclude extraneous material; the papers that did not address the questions put forth in this review or lacked original findings were disregarded at this stage, leaving 28 articles for this study to analyze.

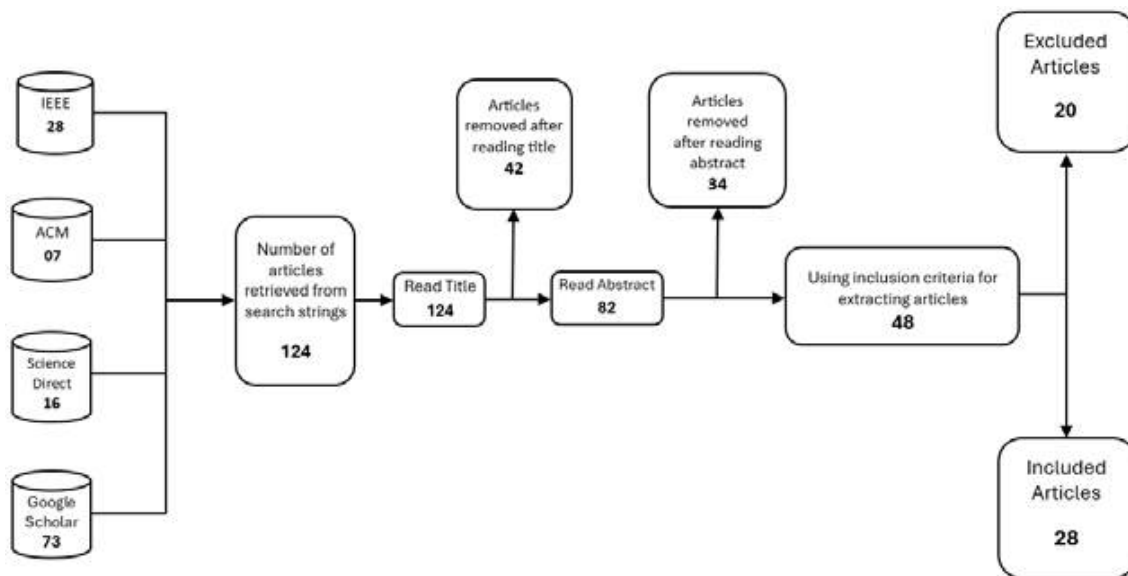


Figure 2. Paper Selection Process

Table 5 presents the summary of the extracted papers. We can observe that most of the covered literature has been acquired from journal publications, 20 out of the 28 covered articles in the review have been published in reputed journals, whereas 8 articles have been published in conference proceedings. The overall coverage of the knowledge sources has been depicted in Fig. 3.

Table 5. Summary of extracted articles

Knowledge Source	Number of Journal papers	Number of Conference papers	Total Number of Articles
IEEE	4	-	4
ACM	2	1	3
Science Direct	5	3	8
Google Scholar	9	4	13

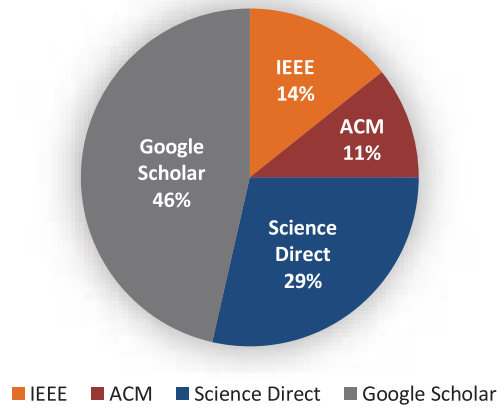


Figure 3. Coverage of knowledge sources

The literature review was performed on research papers published from 2014 to 2024 as shown in Figure 4. From the temporal view of the papers, we can see an increase in research focusing on the use of EDM towards student performance prediction in 2016 and 2021, closely followed by 2023.

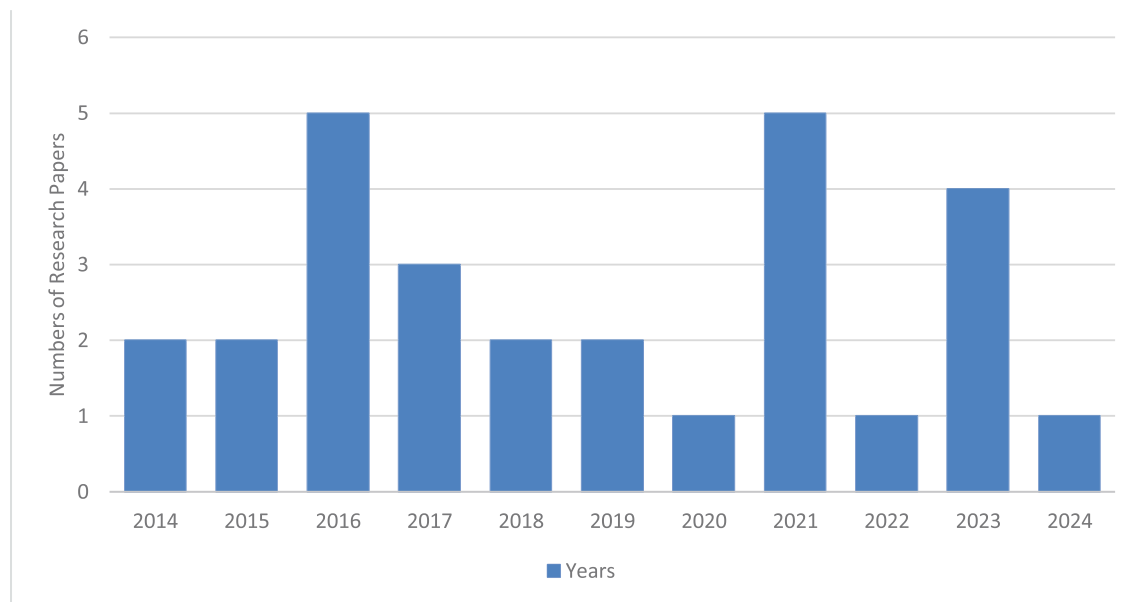


Figure 4. Temporal view of publications

3. Results & Analysis

This section of the literature review presents the findings of the conducted review. The findings have been categorized based on the research questions proposed in section 2.1. We have also attempted to provide a tabular summary of the most used EDM techniques, the features most commonly utilized in the research, and the experimental accuracy achieved using the various EDM classification approaches.

3.1 RQ1: What are the most essential factors explored when forecasting student performance?

Various factors(features/attributes) have been explored in the reviewed research for predicting student performance with the cumulative grade point average (CGPA) being the most frequently used [9], [10], [11], [12], [13], [14], [15]. A primary reason why most researchers use CGPA could be because it has a concrete value that can be used to address future academic mobility; CGPA shows the exact output variable as compared to other elements, therefore, it is the most influential in deciding the survival of students in their studies, and whether they can finish their course. Apart from CGPA, another important feature considered for student performance prediction was internal assessment. Internal assessment estimates student's academic performance through various means such as quizzes, class participation, attendance, assignments, and term grades. According to some studies, the timely analysis of student performance in these activities plays a major role in student overall performance in the course [2], [10], [12], [14], [15], [16]. Also, analysis of these factors can help generate early warning systems and intervention channels which can help the instructor monitor and improve the performance of students in their course [17], [18]. The influence of the previous semester (external assessment) on the outcome of the current semester is a dynamic interplay of academic performance, and knowledge retention [12], [13]. Prequalification ensures that applicants to the university have met certain academic requirements as it suggests that they possess the

fundamental knowledge and skills necessary to succeed in courses at the university level; this can have a positive effect on student performance [13].

Apart from features pertaining to academics or learning, studies have also taken into consideration factors such as demographic information including age, nationality, and gender, which may affect student performance [12], [15]. The demographics of age and gender are frequently explored for performance prediction as they are viewed as inner elements of an individual, which are easy to characterize and quantify. Studies have contradictory findings on the significance of gender in student performance prediction. However, it has been seen that female, and male students follow different learning patterns [19].

Some studies also provide an analysis that parents’ educational background, job, family structure, and family income influence the performance of a student. However, this must be kept in mind that all these comprehensive factors contribute to qualitative analysis of the student's performance [9]. Research has focused on extracting and examining factors from both face-to-face as well as online forms of learning. Some researchers have also obtained additional factors using surveys or questionnaires. However, most of the research has focused on the use of features obtained from physical or face-to-face forms of education. The most common factors explored in the reviewed papers have been categorized in Table 6.

Table 6. Factors explored when forecasting student performance

Factor	Description	Reference
Internal Assessment	Student performance evaluation through quizzes, class participation, attendance, and term results, shaping overall aggregate.	[3], [6], [10], [11], [12], [14], [15], [16], [20], [21], [23], [24], [26], [29]
Prequalification	Ensuring the students meet academic pre-requirements before university enrollment, indicating ease towards university-level courses.	[6], [13], [15], [16], [24], [25], [26]
External Assessment	Performance of the previous semester or academic year influencing performance in the current semester.	[12], [13], [16], [20], [26], [27]
CGPA	A grade point average value indicating overall score summarizing the academic performance of all courses of the semester, year, or academic duration.	[1], [9], [10], [11], [12], [13], [14], [15], [20], [21], [23], [26], [28]
Demographic Information	Considering age, nationality, ethnicity, and gender of students for characterization, identifying potential learning patterns.	[12], [15], [18], [20], [21], [24], [25]
Parental Background	Educational background, family structure, and family income creates an impact on student performance, contributing to a qualitative academic understanding.	[9], [12], [25], [27]

Aptitude Score	A quantitative metric to identify student's potential for success in upcoming academia, mostly used in the institute's entrance testing and assessments.	[1], [4]
E-Learning	Data obtained through a learning management or e-learning system	[4], [9], [16], [22]

3.2 RQ2: Which classification techniques or algorithms for predicting student performance are commonly used in EDM research?

To better understand student learning patterns and design effective pedagogical policies, EDM has been utilized to predict various facets of student academic performance. Regression models have been used to estimate continuous values such as final percentage [7], whereas classification, a supervised machine learning technique, is often considered the better fit for predicting student academic success in terms of performance. Classification models have been used to predict student performance outcomes in a course (pass/fail) [2], and even in a degree program (pass/fail or achieved grade) [6]. Common EDM algorithms are decision trees that create tree-like structures for easy interpretation [3]; Random Forests, an ensemble machine learning approach that combines multiple decision trees to reduce overfitting [2], [6]; Naive Bayes, which works on the probability of features belonging to each class [10]; KNN, is a non-parametric and instance-based learning algorithm [20]; and Multi-layer Perceptron (MLP), a powerful architecture for artificial neural networks that can deal with complex data relationships [16]. These algorithms facilitate the prediction of student academic results, help identify learning patterns, and classify student's performance based on student data.

3.2.1 Naïve Bayes

Naive Bayes is a probabilistic algorithm used for classification. The algorithm determines the probability of a hypothesis given the evidence and selects the class with the maximum probability to classify input data. In the reviewed literature, sixteen studies have utilized the Naïve Bayes techniques to predict student performance. Analysis of their findings has been provided in Table 7.

Table 7. Review Summary - Naïve Bayes

Classifier	Reference	Elements/Features	Accuracy
Naïve Bayes	[6]	Internal assessment, school and college results	83.65%
	[9]	CGPA, parental survey, absent days, resources used, participation in discussion	76.05%
	[10]	CGPA	73.60%
	[12]	CGPA, internal assessment, external, student demographic, parents educational background	75.00%
	[13]	CGPA, pre-qualification, previous semester	73.50%
	[14]	CGPA, internal assessment, lab marks, attendance marks	61.00%
	[15]	CGPA, student demographic, pre-qualification	86.20%

	[16]	Student demographic, pre-qualification, Institutional assessment	67.60%
	[19]	GPA, gender, ethnicity, financial status	67.00%
	[20]	CGPA, assessments, student demographic	73.00%
	[21]	CGPA, internal assessment, student demographic, parents' educational background	86.00%
	[23]	Internal assessment	71.30%
	[24]	Internal assessment, enrolment data, school and college results, demographics	79.72%
	[25]	Student demographics, parents' educational background, qualification, travel time, scholarship	69.70%
	[27]	External assessment, father qualification, social features	91.79%
	[29]	Student performance data	74.00%

3.2.2 K-Nearest neighbors

Among all data mining techniques, KNN is a simple, non-parametric, and instance-based learning algorithm preferred to perform classification and regression. Although this classifier is deemed lazy as it does not readily generate a classification model but rather scans the incoming instances at runtime to make classification decisions, it is still widely adopted as it does not make any prior assumptions regarding the data it has to classify. KNN has been used in eleven of the reviewed studies to analyze and predict student performance as depicted in Table 8.

Table 8. Review Summary - KNN

Classifier	Reference	Elements/Features	Accuracy
KNN	[3]	Internal assessment marks, final exam marks	83.00%
	[4]	Spatial reasoning test, critical thinking, online course data	77.80%
	[5]	Semester grades, gender, age, parents' education	92.25%
	[6]	Internal assessment, school and college results	74.00%
	[9]	CGPA, parental survey, absent days, resources used, participation in discussion	76.70%
	[12]	CGPA, internal assessment, external, student demographic, parents educational background	83.00%
	[16]	Student demographic, pre-qualification, Institutional assessment	82.00%
	[20]	CGPA, assessments, student demographic	74.00%
	[23]	Internal assessment	69.90%
	[24]	Internal assessment, enrolment data, school and college results, demographics	76.28%
	[27]	External assessment, father qualification, social features	88.86%

3.2.3 Decision tree

The decision tree is a non-linear, predictive modeling approach used in machine learning and data mining. The algorithm based on this approach involves recursively partitioning the data hinged on the most informative attributes to create subsets with homogeneous target values [13]. To predict a student's performance, eighteen of the reviewed papers have employed the decision tree method. An analysis of their findings has been listed in Table 9.

Table 9. Review Summary - Decision Tree

Classifier	Reference	Elements/Features	Accuracy
Decision Tree	[1]	Test scores, general aptitude test score, GPA, school GPA, gender.	87.17%
	[4]	Spatial reasoning test, critical thinking, online course data	87.60%
	[5]	Semester grades, gender, age, parents' education	94.55%
	[6]	Internal assessment, school and college results	71.15%
	[9]	CGPA, parental survey, absent days, resources used, participation in discussion	95.20%
	[10]	CGPA	75.90%
	[11]	CGPA, internal assessment	55.50%
	[12]	CGPA, internal assessment, external, student demographic, parents educational background	88.00%
	[14]	CGPA, internal assessment	56.10%
	[16]	Student demographic, pre-qualification, Institutional assessment	72.30%
	[18]	Student Demographic	70.80%
	[19]	GPA, gender, ethnicity, financial status	68.80%
	[23]	Internal assessment	74.60%
	[24]	Internal assessment, enrolment data, school and college results, demographics	80.75%
	[25]	Student demographics, parents' educational background, qualification, travel time, scholarship	73.20%
	[26]	Internal assessment, school and college score, education attainment test, CGPA	80.00%
	[27]	External assessment, father qualification, social features	92.96%
	[28]	CGPA, social features, external assessment	67.37%

3.2.4 Multi-layer perceptron

The classification model constructed by MLP takes the form of an artificial neural network. Such models have demonstrated to be effective in a wide range of tasks, including classification,

regression, and other pattern recognition. It is a popular neural networks architecture used to estimate student’s academic performance [21]. An analysis of the findings of the reviewed papers based on MLP has been provided in Table 10.

Table 10. Review Summary - MLP

Classifier	Reference	Elements/Features	Accuracy
MLP	[6]	Internal assessment, school and college results	71.15%
	[9]	CGPA, parental survey, absent days, resources used, participation in discussion	94.50%
	[15]	CGPA, Student demographic, pre-qualification	81.30%
	[16]	Student demographic, pre-qualification, Institutional assessment	78.70%
	[20]	CGPA, assessments, student demographic	50.00%
	[21]	CGPA, internal assessment, student demographic, parents' educational background	82.70%
	[23]	Internal assessment	74.60%
	[25]	Student demographics, parents' educational background, qualification, travel time, scholarship	69.30%

Figure 5 depicts the frequency of the classifiers’ explored in the reviewed literature. It can be observed that decision tree along with Naïve Bayes are the more preferred classifiers explored in the reviewed literature.

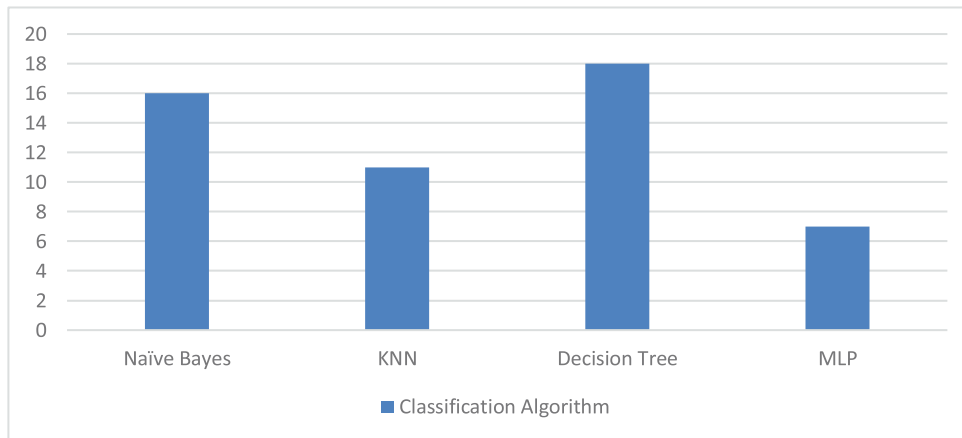


Figure 5. Frequency of classifiers’ explored

3.3 RQ3: What are the limitations or challenges in using EDM for predicting student performance?

While EDM can be a useful tool for instructors and institutions for predicting student performance, it does have some limitations and difficulties. Some of the highlighted

challenges in the reviewed literature include:

1. Quality and availability of the data: The accuracy of forecasts vigorously depends upon the quality and amount of the information accessible. On the off chance that the information is missing, inadequate, mistaken, or one-sided, it can prompt temperamental prediction and frustrate the adequacy of the EDM model [22].
2. Privacy and ethical concerns: Educational data frequently contains delicate information about students. Analyzing and sharing students' information should be carried out delicately, while strictly complying with severe security guidelines and guaranteeing that no singular student information is compromised [7].
3. Interpretability: Some EDM methods, particularly complex AI models, could need interpretability. Understanding why a model makes a particular prediction can be difficult, and at times impossible. This can be worrisome in educational analysis where straightforwardness and an understanding of the classification logic are fundamental for pursuing informed choices [24].
4. Logical Elements: Predicting students' execution is affected by different relevant elements, for example, financial status, parental expectations, gender comparisons, family foundation, and outer impacts. EDM models could battle to catch the intricacy of these elements [23].
5. Data Preprocessing: Frequently, extensive data preprocessing and feature engineering are required when preparing educational data for analysis. Time-consuming and requiring domain expertise are the requirements for cleaning, transforming, and selecting relevant features [2].

4. Discussion

This meta-analysis relies on the efficiency of the EDM techniques (measured using prediction accuracy) as well as the critically important factors that could affect students' performance. Taking into consideration the accuracy achieved by the models generated in the reviewed papers, the overall highest accuracy achieved by the model generated by Naïve Bayes was 91.79% [27], the model based on the decision tree recorded the highest accuracy of 95.20% [9], for the model based on MLP, the accuracy was 94.50% [9], and the model based on KNN exhibited the highest accuracy of 92.25% [5].

As observed, different researchers have investigated various blends of educational elements or features to accomplish better accuracy of their generated models. For instance, when working with the features of CGPA, internal assessment, external scores, student demographic information, and parents' educational backgrounds, the model based on Naïve Bayes achieved an accuracy of 75%. Working with the same data and features, the model based on KNN achieved an accuracy of 83%, whereas the decision tree outperformed working on the same data with an accuracy of 88% [12]. Another study used student demographic data, pre-qualification information, and institutional assessments to predict student performance using the approaches of MLP, KNN, Naïve Bayes, and decision tree; with the model based on KNN outperforming with an accuracy of 82% [16]. When working with the features of internal assessment focusing on marks obtained in

subjects being studied, school and college results, the model based on the decision tree and MLP achieved an accuracy of 71.15%, while the model based on KNN achieved an accuracy of 74%, and the model based on Naïve Bayes had an accuracy of 83.65% [6]. The difference in classifier performance signifies that there is no ‘one fit all’ approach when it comes to using classification algorithms. Classifier performance greatly depends on the size and quality of the data as well as the features being considered [6], [9], [24].

Based on the reviewed literature, and the uncovered challenges and limitations, some areas that need attention, possible solutions to the identified challenges, and potential future research directions can be categorized as:

Identification of Factors and Interpretability: The main motivation behind the field of EDM is the discovery of patterns and facets of student learning that make a student perform well or poorly in a course, semester, or across a degree program and then use these discovered patterns to fashion, among others, better learning infrastructures, early warning systems, and intervention mechanisms. In order to accomplish this, it is essential to not just focus on using varied classification algorithms to predict student academic achievement but rather take the research process a step further and try to identify the features that most influence student performance. The uncovered factors should then be communicated to the administrators and instructors so that, where required, interventions can be timely planned and pedagogical policies be reshaped to add quality to the system of education. While several of the reviewed papers have undertaken this exploration and provided conclusions based on factors that help a student excel or hinder academic achievement [6], [9], [10], [24], many papers have solely focused on a comparative analysis of classification performance, limiting their research to simply finding the best performing classifier [1], [3], [4], [13], [14], [15], [17], [18], [19], [27], [29]. Also, although Naïve Bayes has been utilized in the majority of studies, in most studies where both the approaches of Naïve Bayes and decision trees have been utilized, we can observe that the model based on the decision tree has almost consistently outperformed [9], [10], [12], [16], [19], [23], [24], [25], [27]. A benefit of using the decision tree approach is the resulting interpretability of the model [6], [24]. Due to the generated tree-based model, educators can readily understand the prediction logic and can use the logic to create early warning and intervention systems.

Feature Selection: Predictive models such as decision trees, Naïve Bayes, KNN, and MLP offer a promising roadmap for predicting student performance and improving results. Nonetheless, they are not without their limitations and difficulties. The accuracy and dependability of these models vigorously rely upon the nature of the information, making information quality a significant concern [6]. Researchers analyzing, exploring, and experimenting on educational data need to understand the significant impact of low quality, imbalanced, small datasets on their final results. EDM is a data driven field, and as such low quality data will always result in low quality results. Additionally, choosing the right arrangement of elements is fundamental, as superfluous, or inadequately picked highlights can obstruct accuracy. Few of the reviewed studies explored feature selection approaches in their experiments [1], [6], [7], [24]. The use of feature selection approaches such as t-distributed Stochastic Neighbor Embedding [1], Attribute Evaluator Wrapper with Best First Search [7], Filter with Ranker search [7], and Correlation-based Feature Subset Selection Evaluator [24] is recommended to ensure only features that play a part in the final prediction of student performance are used in the experiments. This will ensure the integrity of the results.

Data Imbalance: Overfitting can likewise be an issue, prompting unfortunate speculation of inconspicuous information. This can again cause biased results towards the majority class. A limited number of studies have focused on balancing student data before the generation of the predictive models [7], [9], [14], [24]. The application of oversampling using the Synthetic Minority Oversampling Technique (SMOTE) has shown promising results in generating synthetic samples of the minority class, thus reducing the imbalance between the majority and minority classes. The comparative findings between the use of balanced and unbalanced data further iterates the need for researchers to conduct a proper statistical analysis of the data at their disposal and ensure that the classes are balanced before moving on to the classification process [24].

Availability, Privacy, and Ethical Concerns: As far as the availability, privacy, and ethical concerns regarding student data, with the increased interest in exploring educational data, educational institutes have implemented stringent policies regarding the access and use of student data. Researchers are allowed to collect and experiment with this data after a thorough ethical review board allows them access to the data ensuring that the data is anonymized before it is used and published [6], [10], [24]. Reputed journals do not proceed with publication unless researchers provide ethical board approvals. This ensures the privacy of student data. Moral contemplations are central while managing delicate instructive information to guarantee understudy security and information morals.

Model Optimization, Ensemble, and Hybrid Models: Hyper-parameter tuning or model optimization is an issue that has not been the focus of a majority of the covered research. The same goes for performing classification using ensemble approaches. Although promising results have been uncovered using ensemble methods, this remains a less explored area [9], [24]. Also, during the COVID pandemic, educational institutes, to a large extent, shifted to an online format of education causing a surge in the generation of e-learning and online heuristic data [5]. Although many institutes have switched back to a face-to-face form of education, they supplement the use of online sources for assignments, tests, and notes. This has also opened avenues for research focusing on additional factors. Features such as course resources accessed, participation in forums or class discussion groups, number of interactions, time of joining a class, attendance, assignment or homework submission record, and a myriad of other factors that were previously unavailable can now be used in EDM based research. Hybrid datasets containing data compiled from physical as well as online mediums have the potential to shape an in-depth understanding of student learning patterns. The increase in the number of parameters to explore along with the increase in the size of the data can also allow exploration of EDM to move from tradition machine learning to ensemble and deep learning approaches.

An interesting observation during the review was the reliance on models must be regularly updated to remain relevant because educational environments are always changing. Also, it needs to be underscored that educational data is hierarchical in nature; data at different levels correlates and affects the overall picture. Ultimately, an absence of space mastery and powerful mediation systems can restrict the commonsense accuracy of these models.

5. Conclusion

This paper reviewed research between 2014 and 2024 on using various classification techniques to

predict students' performance. Most researchers have utilized internal assessments such as assignments, lab work, quizzes, class participation, attendance, assignments, terms grades, and CGPA as their primary elements. The most common techniques in EDM for predicting students' performance are Naïve Bayes and decision trees. Out of the reviewed papers, eighteen papers used decision tree techniques to predict student performance, evaluating the model performance on the metric of accuracy. We can thus conclude that classifiers based on the decision tree are popular and preferred classification methods in EDM research. The decision tree is followed by the Naïve Bayes classifier as the second most used approach. However, the results of the studies based on decision trees have constantly outperformed, and with the advantage of generating a readily interpretable model, the decision tree appears to be an attractive approach for student performance prediction. Further work in this area of research can focus on the utilization of feature selection and data balancing techniques and how their use influences the overall classification outcome. Although there are still concerns and difficulties in the implementation of EDM, it is a useful asset for recognizing patterns, acquiring knowledge, and giving early mediation to further understand student learning behavior. By tending these limitations and using EDM, instructors can leverage data-driven approaches to enhance the learning experience and academic performance of their students.


References

1. Alhazmi, E., &Sheneamer, A. (2023). Early predicting of students performance in higher education. *IEEE Access*, 11, 27579-27589.
2. Abu Saa, A., Al-Emran, M., &Shaalan, K. (2019). Factors affecting students' performance in higher education: a systematic review of predictive data mining techniques. *Technology, Knowledge and Learning*, 24, 567-598.
3. Karthikeyan, K., &Kavipriya, P. (2017). On improving student performance prediction in education systems using enhanced data mining techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, 7(5).
4. Çetinkaya, A., Baykan, Ö. K., & Kirgiz, H. (2023). Analysis of Machine Learning Classification Approaches for Predicting Students' Programming Aptitude. *Sustainability*, 15(17), 12917.
5. Daligcon, A. G., Priyadarshini, J., &Decena, L. R. (2024). Unveiling the Best-fit Model: A Comparative Analysis of Classification Methods in Predicting Student Success. *International Journal of Information Technology, Research and Applications*, 3(1), 12-19.
6. Asif, R., Merceron, A., Ali, S. A., &Haider, N. G. (2017). Analyzing undergraduate students' performance using educational data mining. *Computers & education*, 113, 177-194.
7. Bujang, S. D. A., Selamat, A., Ibrahim, R., Krejcar, O., Herrera-Viedma, E., Fujita, H., & Ghani, N. A. M. (2021). Multiclass prediction model for student grade prediction using machine learning. *IEEE Access*, 9, 95608-95621.
8. Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering—a systematic literature review. *Information and software technology*, 51(1), 7-15.
9. Sahlaoui, H., Nayyar, A., Agoujil, S., &Jaber, M. M. (2021). Predicting and interpreting student performance using ensemble models and shapley additive explanations. *IEEE Access*, 9, 152688-152703.
10. Mengash, H. A. (2020). Using data mining techniques to predict student performance to support decision making in university admission systems. *IEEE Access*, 8, 55462-55470.
11. Mehboob, B., Liaqat, R. M., &Saqib, N. A. (2016). Predicting student performance and risk analysis by using data mining approach. *International Journal of Computer Science and Information Security (IJCSIS)*, 14(7), 69-76.
12. Ashraf, A., Anwer, S., & Khan, M. G. (2018). A Comparative study of predicting student's

- performance by use of data mining techniques. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, 44(1), 122-136.
13. Shruthi, P., & Chaitra, B. P. (2016). Student performance prediction in education sector using data mining. *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(3), 212–218.
 14. Jishan, S. T., Rashu, R. I., Haque, N., & Rahman, R. M. (2015). Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique. *Decision Analytics*, 2, 1-25.
 15. Pavithra, A., & Dhanaraj, S. (2018). Prediction Accuracy on Academic Performance of Students Using Different Data Mining Algorithms with Influencing Factors. *International Journal of Scientific Research & Management Studies*, 7(5).
 16. Rafique, A., Khan, M. S., Jamal, M. H., Tasadduq, M., Rustam, F., Lee, E., ... & Ashraf, I. (2021). Integrating learning analytics and collaborative learning for improving student's academic performance. *IEEE Access*, 9, 167812-167826.
 17. Durairaj, M., & Vijitha, C. (2014). Educational data mining for prediction of student performance using clustering algorithms. *International Journal of Computer Science and Information Technologies*, 5(4), 5987-5991.
 18. Khudhur, M. E., Ahmed, M. S., & Maher, S. M. (2021). Prediction of the Academic Achievement of Pupils Using Data Mining Techniques. *Webology*, 18(2), 1355-1364.
 19. Ahmad, F., Ismail, N. H., & Aziz, A. A. (2015). The prediction of students' academic performance using classification data mining techniques. *Applied mathematical sciences*, 9(129), 6415-6426.
 20. Zohair, A., & Mahmoud, L. (2019). Prediction of Student's performance by modelling small dataset size. *International Journal of Educational Technology in Higher Education*, 16(1), 1-18.
 21. Mueen, A., Zafar, B., & Manzoor, U. (2016). Modeling and predicting students' academic performance using data mining techniques. *International Journal of Modern Education and Computer Science*, 8(11), 36.
 22. Wong, J., Khalil, M., Baars, M., de Koning, B. B., & Paas, F. (2019). Exploring sequences of learner activities in relation to self-regulated learning in a massive open online course. *Computers & Education*, 140, 103595.
 23. Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, 9(1), 11.
 24. Meghji, A. F., Mahoto, N. A., Asiri, Y., Alshahrani, H., Sulaiman, A., & Shaikh, A. (2023). Early detection of student degree-level academic performance using educational data mining. *PeerJ Computer Science*, 9, e1294.
 25. Osmanbegović, E., Suljić, M., & Agić, H. (2014). Determining dominant factor for students performance prediction by using data mining classification algorithms. *Tranzicija*, 16(34), 147-158.
 26. Altujjar, Y., Altamimi, W., Al-Turaiki, I., & Al-Razgan, M. (2016). Predicting critical courses affecting students performance: a case study. *Procedia Computer Science*, 82, 65-71.
 27. Mousa, H., & Maghari, A. (2017). School student's performance prediction using data mining classification. *International Journal of Advanced Research in Computer and Communication Engineering*, 6(8), 136-141.
 28. Singh, W., & Kaur, P. (2016). Comparative Analysis of Classification Techniques for Predicting Computer Engineering Students' Academic Performance. *International Journal of Advanced Research in Computer Science*, 7(6).
 29. Pallathadka, H., Wenda, A., Ramirez-Asís, E., Asís-López, M., Flores-Albornoz, J., & Phasinam, K. (2023). Classification and prediction of student performance data using various machine learning algorithms. *Materials today: proceedings*, 80, 3782-3785.

A Hybrid Model for Human Behavior Recognition Using Emotions, Sentiments, and Mood Features

Asia Samreen^{a*}

 ^aDepartment of Computer Science, Bahria University , Karachi, Pakistan
<https://orcid.org/0000-0002-1220-2959>, asiasamreen.bukc@bahria.edu.pk

Syed Asif Ali^b

^bDepartment of Computer Science, Sindh Madressatul Islam University, Pakistan
aasyed@smiu.edu.pk

Hina Shakir^c, Muhammad Hussain^c

^cDepartment of Software Engineering, Bahria University , Karachi, Pakistan
hinashakir.bukc@bahria.edu.pk, Engr.m.hussain.bukc@bahria.edu.pk

*Corresponding Author: Asia Samreen asiasamreen.bukc@bahria.edu.pk

Abstract

While social networking is a powerful communication tool, the obscured behavior of individuals on social networks remains a significant problem for users. Currently, research work is being focused on formulating mechanisms to determine the obscured behavior of users for secure and trustworthy social media. The proposed model employs mathematical formulation and multinomial classification of mood and emotions to analyze the conduct of an individual, thus enabling social trust on social media. First, natural language processing techniques are applied to predict the emotions, moods, and sentiments of an individual from the text, and then a mathematical model is applied to gather a comprehensive picture of one's behavior using calculations at numerous instants. Finally, a subsequent trust state log is built in terms of positive and negative states which show the devotion in behavior in terms of mood, sentiments, and more significantly emotions. The efficiency of the proposed work has been demonstrated using simulation-based and real-world datasets along with individual behavior graphs for various conversations.

Keywords: Social Networks, Social Trust, Positive and Negative state log, Mathematical framework, Hybrid tactic, Multinomial classification, Human behavior, Natural Language Processing

1. Introduction

A precise definition of Human behavior is “Human behavior is a dynamic interplay of three constituents in scientific research: attitudes, cognition, and emotions”, which shows human actions and feelings play a great role in developing behavior and a cultural or environmental element causes the situation of some specific behavior. Understanding human behavior could help predict

an individual's positivity or negativity. To estimate behavior mathematically, it is required to convert behavior-building elements such as mood, feelings, and emotions into numeric form. For this purpose, words can be categorized first as positive, negative, or neutral, and then a polarity score can be used to calculate human behavior. Social network users participate in various conversations daily. These conversations can play an important role in improving or deteriorating the mental health of individuals [1]. Therefore, it is necessary to understand the components of human behavior to extract patterns from their daily activities or actions [2]. Every human being has certain feelings, moods, sentiments, and emotions, and all these components play a vital role in building human behavior.

Despite recent advances in predicting human behaviors based on textual information, there is a dearth of major advances in data analytics systems for identifying, determining interrelationships, and forecasting human cognitive, emotional, and social behaviors [3].

In the quest for human behavior prediction on social networks (SN) which is constantly evolving, building trust has proven to be a challenging issue [4]. Social trust is not only an individual property but also a property of society [5], and it represents expectations of strangers' general cooperation and effectiveness while networking. The hidden behavior of SN users poses a challenge in maintaining social trust. The proposed research findings prove that "words" can be used to predict and analyze the polarity of human thoughts if certain patterns in textual data of a conversation could be computationally identified. In the presented work, the conduct or behavior of an individual on social networks is computed and a mechanism to maintain social trust in social networks is devised.

1.1 Factors that Influence the behaviour

In this section, factors that can significantly influence a person's behavior are discussed. A thorough understanding of such factors can enable behavior prediction of an active user with tag words typed by him/her on a social network such as Twitter or Facebook etc. Attitude is closely related to behavior as it can be defined as the tendency of perception to give a positive or negative response. A connection between behavior and attitude can be understood by "privacy paradox", a term used to explain people who often assert that they care about privacy on online systems, but their actions do not reflect their concerns [6]. A thorough study of various theories about the attitude-behavior relationships led Ajzan et al. to conclude that the aforementioned relationship might show inconsistent results with varying situations, [7]. It has been observed that individuals often hide their actual personalities to portray the best possible behavior on social networks. While physical environments have several constraints, virtual environments are prone to various communal threats such as irrational behavior, sudden changes in behavior, etc. Thus, unethical behavior on social networks cannot be explained accurately.

Intention is another factor that can be represented as the cognitive state of doing some action. However, the translation of intention into action has been an active area of research for the past few years. Behavioral intentions are self-instructions to perform specific actions to achieve desired outcomes [8]. People frequently share their intentions on social media, such as "I might go to the party at 6 PM," "I'm going abroad tomorrow," and so on, but these intentions do not provide a clear picture of the individual's thought polarity. Though intention is an important factor in understanding the current state of a person, it cannot help to estimate the action that can be performed by the subject. On social networks, conversation or communication is mainly performed via writing words, sentences, or symbols representing emotions. For certain systems, sentiment

analysis might be insufficient and hence require emotion detection, which precisely determines an individual's mental state [9].

A human sentiment is a thought, perspective, proposition, or attitude about some state of affairs or precisely about a situation [10]. Popular techniques, including word embedding, can play a vital role in extracting information from sentiments hidden in some text [11]. Another important factor is mood, which is the state of a person and is less intensive as compared to feelings, emotions, and sentiments. Mood is an integrated and widespread affective reaction that is believed to influence cognition and behavior [12]. Adopting the concept of reinforcement learning, mood can be taken as an advantage of some action [13]. Mood and emotions are correlative, and if mood is correctly found, then, by applying intelligent techniques, emotions can also be detected with high precision [14].

1.2 NLP Techniques for Behaviour Analysis

Analyzing people's emotions through their writings is an emerging trend in NLP research [15]. Text-based data has now become a significant source of information regarding the relationships, links, and behaviors of individuals and groups. There are various methods to extract meanings from the text such as Neural Networks to find political partiality [16]. NLP techniques use numbers associated with words, and the statistical analysis of text then helps to predict patterns based on past behavior.

NLP-based techniques such as term frequency, inverse document frequency (TF-IDF), part of speech tagging (POS), etc., and prediction using machine learning provide better results for large datasets. Naive Bayes has been applied in the proposed work which is a classification approach that uses Bayes' theorem to calculate the likelihood of a given feature vector being linked with a label. Logistic regression is a linear classification approach that learns the likelihood of a sample belonging to a specific class. Logistic regression seeks the optimal decision boundary that optimally divides the classes. The logistical function, also known as the sigmoid function, is used to construct the logistic model, in which values range from negative infinity to positive infinity and the output is between 0 and 1. Both approaches are used classification and prediction of some classes. Furthermore, for the proposed work both approaches are used to extract or predict the emotions, moods, and sentiments from the test data.

The ongoing research is directed at uncovering the hidden semantics to predict intent. The objective of sentiment analysis is to classify the text's polarity like conflict, neutral, positive, and negative [17]. Therefore, human behavior can be modeled using a mathematical formulation that depends upon the polarity of an individual's feelings, emotions, and mood. Hence, if NLP is combined with mathematical models, then behavior recognition with a certain accuracy can be obtained.

1.3 Contribution of the proposed research

The presented research work contributes a novel hybrid model to detect human behavior and employs a mathematical model combined with natural language processing. The previous research studies [2]; [18]; [19] provided either theoretical backgrounds or learning methods to describe human behavior. Furthermore, [9] presents a statistical approach for the classification of text that is categorized as human behavior such as news articles or product data. Furthermore, Our proposed model takes emotions, moods, and sentiments as features and estimates behavior using

mathematical formulation. The study findings show that human behavior can be modeled as a function of the mood, sentiments, and feelings of individuals, along with the environment and special events. Emotions' classification and predation provide the polarity of emotions and feelings. This research aims to go a step ahead and find the state of polarity in the behavior of an individual using the proposed mathematical model. In the presented work, textual data has proven to be a good tool to reveal divergence of thought and can be used to enable trust levels on social networks.

The paper has been organized as follows. Section 2 describes some techniques used to assess human behavior, especially for textual data. Section 3 explains the proposed model, while Section 4 provides both simulation-based and real-data-based results. Section 5 describes the conclusion and future work.

2. Literature Review

Spoken words are a great source to evaluate the emotions or moods of some individuals and the mapping of moods or emotions can give certain outcomes like frustration, happiness, etc [20]. For a linguistics-based automated system, words play an important role as they not only exhibit the continuity of actions but also reveal different aspects of human behavior. The language (combination of words) is shown to be used for evil purposes, such as wars, by humans [21]. The significance of words becomes the backbone for such problems as emotion detection, personality classification, mood-based sentiment analysis, and behavior prediction by analyzing the hidden meaning of words. Twitter, for example, is a well-known social network where millions of users share or express their feelings by leaving comments, and by analyzing those comments, personality can be judged or intentions can be predicted. Putting feelings into words, or "affect labeling", can attenuate our emotional experiences [22].

Applying behavioral theories as features in combination with machine learning yields the best results [23]. For the last decade, people have been communicating via social networks, and, therefore, online communication has become a great avenue of research [19]. Emotions significantly influence action, learning, and perception, but their role remains debated [20]. The authors in [14] describe that if a classical algorithm such as a neural network classifier is combined with an evolutionary algorithm like a genetic algorithm (GA), then it can produce better results. [24] gave a demonstration of a conceptual model that would detect risk by taking the user's observable actions into account. It was claimed by [25] that various human behaviors can be predicted accurately by employing dynamic models [25]. If the communication is virtual or on social networks, especially textual conversation, then it becomes more difficult to identify the malicious activities. By predicting the future activities of an individual, it is possible to mitigate the risk of fraud and stop an individual from attempting dangerous actions like suicide. Therefore, a time-based evaluation is required.

The question of how to accurately predict human behavior has yet to be answered [18]. There are various intelligent systems based on machine learning developed to provide services to the community [26]. However, designing human-centered intelligent systems aims to develop human interaction-based systems with secure communication [18]. With the increased use of social networks, acquainting oneself with new people has become difficult, as in various reported cases, a factor of fraud or unexpected behavior has been observed. Such disorders in the behavior of an

individual or group on social networks can be substantially examined through human behavior prediction [3].

Authors in [2] have proposed a probabilistic model to explore the specialty of each persona extracted from the texts and groupings and further investigate personae based on the specialties. In psychology, personality refers to consistencies in a person's behavior across various situations and, over time, how a person generally tends to respond. The diversity of human behavior depends upon various factors like culture, mood, surroundings, etc., which also makes it difficult to construct a concrete model to predict human behavior accurately [18]. Reinforcement learning formalism can be used for better results [20]. Emotions are a crucial aspect of human life, often characterized as complex patterns of reactions. These are the ways through which individuals cope with significant situations and can be communicated in various ways [27]. A summarised view of the adopted techniques is given in Table 1.

Table 1. Inspiration from the existing work

Research paper	Features	ML Techniques	ML +NLP	Psychological Theories
[11]	Words	-		-
[12]	Mood (lexions)	-	*	-
[13]	Mood (lexions)	-	-	*
[14]	Emotions & mood (theoretical concept)	*	-	*
[18]	Behaviour (theoretical concept)	*	-	*
[9]	Behaviour (Text data)	-	*	-
[16]	Classification of Text	-	*	-
[20]	Emotions (lexicons)	-	*	*
[27]	Emotions (lexicons)	-	*	-

3. Proposed Model to Detect Human Behaviour

3.1 Logical description of the proposed model

The proposed model performs a temporal evaluation of texts. Since an initiated conversation is often continued for various time intervals, the proposed model predicts the polarity of sentiment, mood, and emotions for each comment written by some user during time intervals to the conversation. Then, the polarities are summed using mathematical formulations at different compute and evaluate the behavior of each user or ID participating in the conversation. Figure 1 depicts the described model of behavior recognition and evaluation.

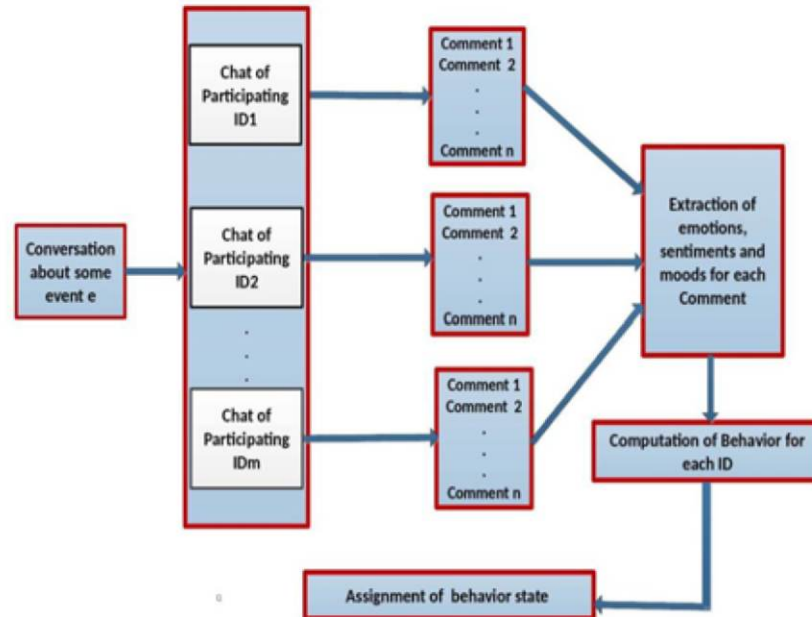


Figure 1. Proposed Model

The theoretical model of the proposed research is inspired by Ajzen [7], which emphasizes that attitude is directly proportional to behavior. In the presented study, while emotions and sentiments of words have been directly calculated, mood has been derived from emotions. A person can have positive or negative sentiments due to his or her perspective of belief [16]. For instance, if someone says, "I don't like this movie because the director of the movie is kind of a flop director," or, "the person is very nice, as I know his family is reputed to be," etc., then it is obvious that negative, neutral or positive sentiments are all influenced by some percentage of belief.

Along with sentiments, emotions E too play an important role in carrying out some intentions or intending to do some actions. "The paper was very difficult (belief) and the result is very scary (negative sentiments), but I scored 80% marks (emotions: happiness, because of the high score). "Intuitively, I was expecting that in a few days everything would be going very well". The above-mentioned example shows a positive action that could be taken by the subject. Emotion lexicons creating a positive sense would make the mood positive, such as "happiness on success," while those that may have a negative sense can negatively impact the mood, such as "gloomy moon." Since the polarity of each emotion has been considered, it is easier to define the mood in a two-dimensional bi-polar space and cluster all those words as positive, negative, mixed, or moderate.

Russell [28] discusses direct circular scaling for words showing emotions shown in Figure 2.

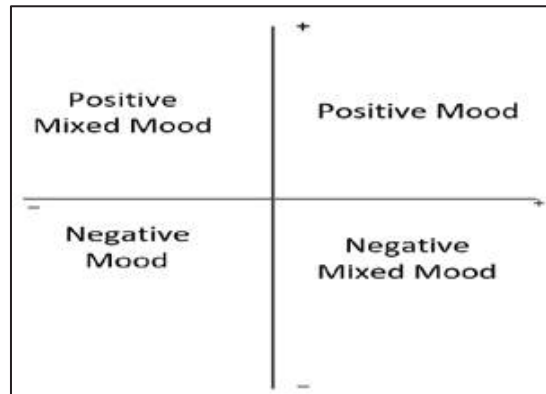


Figure 2. Circular scaling for mood

Few definitions have been furnished below to elaborate the proposed model and to device a mathematical equation for human behavior calculation.

Definition: A conversation is a text-based social engagement regarding a certain topic.

Definition: A human activity is an action at a particular time instant that can be performed by some individual or can cause another event to happen.

Definition: An event is something that will be under discussion at some instant of time

Definition: A situation in a social interaction is a proposal or opinion upon which a response is expected.

Definition: An environmental factor is a feeling that can cause a person to think positively or negatively.

3.2 Mathematical description of proposed model

Consider a conversation C_t initiated at any time instant t for a time interval $[0, T]$, $\forall t \in R^+$ as a series of n words $\sum W_i, \forall i \in Z$ which could be defined as follows:

$$C_t = \sum_{i=1}^n W_i(1)$$

Further, these words can be categorized as useless words or Non-keywords (NKW) and keywords (KW). Hence Eq. (1) can be re-written as:

$$C_t = \sum_{i=1}^n W_i = \sum_{i=1}^n (NKW + KW)(2)$$

As discussed above that behavior is proportional to overall attitude [7]. A relationship among behavior traits such as mood M , emotions E , and sentiments S for human behavior B_{ID} of a user ID from a conversation C_t , has been described as follows:

$$B_{ID}(C_t) \propto (S + M + E)(3)$$

At any instant of time t , a user on social network may have certain sentiments with a percentage or probability of belief. The overall user sentiments in a conversation are represented as given below:

$$S = \sum_{j=1}^n s_j b_j(4)$$

Where s_j denotes the sentiment and b_j shows the belief at time index j . Negative, neutral, or positive sentiments are all mostly influenced by some percentage of belief. The overall emotion E of a user along with the effects of cause on emotions is expressed as:

$$E = \sum_{j=1}^n e_j c_j \quad (5)$$

e_j represents the emotion at any time index j and c_j is the causal effect on emotion e_j . Equation 6 represents the mood of a user at time index j of a conversation, including all mood variances m for environmental factors v , as follows:

$$M = \sum_{j=1}^n m_j v_j(6)$$

Finally, by integrating all of the elements, human behavior can be estimated as given in Equation 7:

$$B_{ID}(C_t) = \Psi(S + M + E)(7)$$

The constant Ψ is termed as event intensity which plays an important role in changing the overall behavior of a user of social network and can be calculated as follows:

$$\Psi = \frac{E_n}{c_t} \times \frac{KW}{c_t}(8)$$

Here E_n represents n occurrences of an event e_v . An event e_v can be repeated several times during a single conversation continued over several intervals of time as often observed on Facebook or other social networks, therefore

$$e_v = \begin{cases} 1 & \text{if } e_v \in TE \\ 0 & \text{otherwise} \end{cases} (9)$$

Here TE represents a set of time-based events. It is a subset of keywords represented as follows:

$$TE \subseteq KW \text{ (set of keywords) } (10)$$

3.3 Implementation Steps

3.3.1 Simulation-based implementation

To perform the task random values were generated and calculations have been done for sentiments, moods, and emotions. Event intensity has also been calculated with the help of random values. Finally, behaviour patterns were generated for random IDs. The values for emotions, sentiments, and moods are calculated using equations 4, 5, and 6 for emotions, sentiments, and moods. Simulations with random numbers have demonstrated the importance of belief for sentiments, cause for emotion, and environmental factors for mood. All these values are generated following equation 11.

$$b_t, c_t, v_t \leq 1 \quad \forall t \in T \quad (11)$$

3.3.2 Real data based simulations

The implementation of proposed human behavior detection model is carried out using the steps of data set preparation, classification of mood and sentiments and finally calculation of user/ID behavior.

3.3.2.1 Data set preparation

Two emotion datasets were used to evaluate the performance of the proposed behavior detection model. Between the two, dataset 1 is a simple dataset, and dataset 2 is a complex dataset that consists of unclear and short sentences. The purpose of using two different datasets is to evaluate the proposed model for all kinds of sentences that are frequently used in conversations. Necessary steps such as conversion of each sentence to small case, tokenization, removing punctuation, special character removal, and removal of web links have been carried out to prepare the dataset for the proposed model implementation. Both datasets were recompiled, adding the moods feature as described above in Figure 2 as well as Tables 2 and 3. All the calculations are based on the parameters mentioned below in Table 3.

Table 2. Mood categorization

Class/ Category	Features based categorization		
Mood	Good	Moderate	Bad
Emotions	Joy, Surprise, Exited	Neutral	Anger, disgust, sadness, fear
Sentiments	Positive	Neutral	Negative

Table 3. Depiction of datasets

Comments	Speaker	Emotion	Sentiment	Mood	Conversation
ive been taking or milligrams or times recommended amount and ive fallen asleep a lot faster but i also feel like so funny	ID6	Surprise	Positive	G	1
i feel like i have to make the suffering i m seeing mean something	ID6	Sadness	Negative	B	1
Really?	ID4	Joy	Positive	G	1

3.3.2.2 Behavior Computation

As aforementioned, the proposed model is a combined approach of both NLP and mathematical computation. The dataset contains about 9 conversations and more than 10000 sentences. The model has been trained for 70% data and the remaining 30% is used to test the system. For multinomial classification and prediction of emotions, moods, and sentiments, the logistic regression (LR) model and the Naive Bayes (NB) approach are used, while mathematical equations are used to compute behavior. The frequency count technique (TFIDF) is employed to find events and event intensity, while keyword finding has also been carried out by using the same. An event's intensity (Ψ) is calculated by using equations 1, 2, and 8. The calculated value is then used to

compute behavior as described in equation 7 for all IDs engaged in conversations on social networks.

3.3.3 Setting the states of individuals

If an individual remains positive throughout a conversation, he or she will be in a Positive (*P*) state whereas if any individual shows a negative attitude then he or she will be in a Negative (*N*) state. Figure 3 depicts state transition at some time instant $t_j \in T$. If this state diagram is recorded for a series of events, it could produce a chain of positive and negative behavior which will be helpful to predict the individual's future intentions.

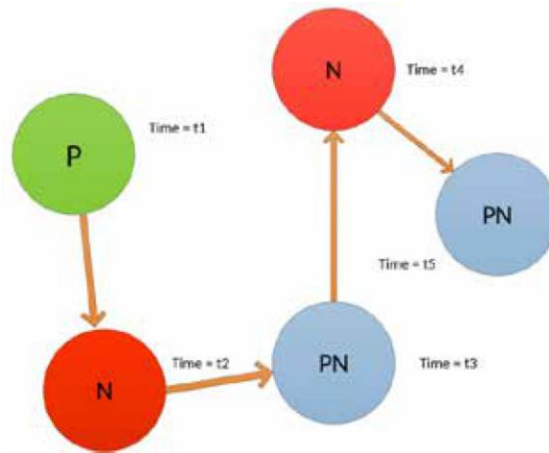


Figure 3. State assignment to IDs

4. Results

4.1 Simulation-based results

For the evaluation of the presented technique on a simulation, random numbers are generated to obtain 100 observations. In Table 4, all the data that is required to estimate sentiments, emotions, and moods has been mentioned. Belief b_t , environment factor v_t , cause c_t , mood m_t , emotion e_t , and sentiments s_t are used to calculate overall Sentiments (*S*), Moods (*M*), and Emotions (*E*).

Table 4. Random values for calculation of behavior state of a sample ID for conversation C_i

Time interval	b_t	c_t	v_t	s_t	m_t	e_t	$S=s_t*b_t$	$M=m_t*v_t$	$E=e_t*c_t$
1	0.105	0.448	0.203	0.105	0.23	0.286	0.011	0.128	0.047
2	0.924	0.192	0.916	1.029	0.203	-0.838	0.951	-0.161	0.186
3	0.976	0.653	0.789	0.053	-0.586	-0.838	0.052	-0.547	-0.462
4	0.624	0.992	0.788	0.0913	0.202	0.39	0.057	0.387	0.159
5	0.86	0.523	0.607	0.0913	0.809	0.663	0.079	0.347	0.491
6	0.907	0.713	0.738	0.653	0.809	-0.09	0.592	-0.064	0.597
7	0.726	0.825	0.226	0.453	0.98	-0.09	0.329	-0.074	0.221

8	0.569	0.788	0.194	0.92	0.615	1.153	0.523	0.909	0.119
9	0.842	0.864	0.887	0.786	0.615	2.057	0.662	1.777	0.546
10	0.702	0.851	0.259	-0.89	0.874	-3.599	-0.625	-3.063	0.226

Random data was generated for total words, keywords, and the total count of event words (Table 5), to simulate text-based data for event intensity calculation. Equations 5, 6, and 7 are used to calculate values for S, E, and M, respectively, and then B is estimated by using equation 8. The aforesaid calculation will demonstrate the complete scenario of the system.

Table 5. Behavior state calculation from textual data for ID , '0'

Conversation	ID	Tot_Words	Key_Words	Event_Count	Psi(Ψ)	Beh_State
1	0	870	653	27	0.023	Positive
2	0	1554	1166	31	.015	Negative
3	0	928	696	23	0.019	Negative
4	0	193	145	27	0.105	Positive
5	0	1421	1066	25	0.013	Negative
6	0	1533	1150	32	0.016	Positive
7	0	570	428	30	0.040	Negative
8	0	326	245	26	0.060	Positive
9	0	1338	1004	33	0.019	Positive
10	0	1892	1419	34	0.013	Negative

To show the graphical results, values for 30 observations have been generated, and for each conversation, a behavior state of each of the 4 IDs has been computed that can show the behavior pattern of that ID. Figure 4 (from A to D) shows the graphical representation of the behavioral states of different users who participated in conversations. The values of emotions, moods, and sentiments along with event intensity(Ψ) are used to calculate behavior state for each conversation. The value of behavior can be either positive or negative which describes the user's behavior for each conversation as the sum of all values. Conversations are represented on the X-axis, while an estimated user's conduct, such as ID1, is represented on the Y-axis. Results show that ID1 usually exhibit the moderate behavior whereas ID2 shows the negative behavior. While ID3 shows great change for some interval of time which is natural, ID4 remains positive in almost all conversations.

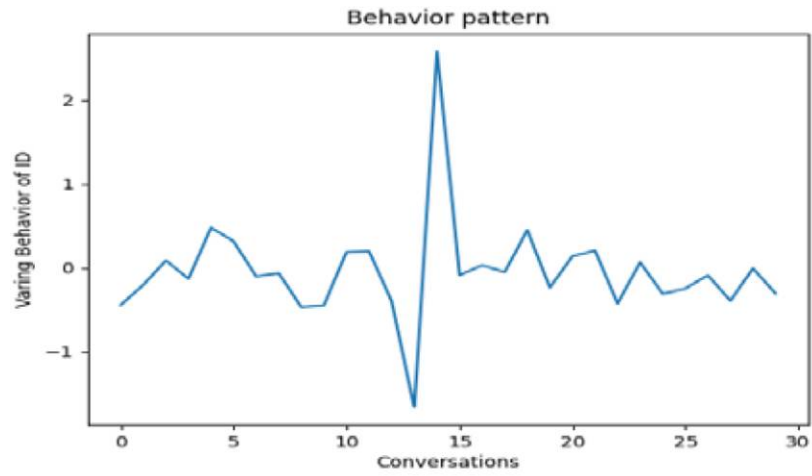


Figure 4(A). Behaviour pattern of ID₁

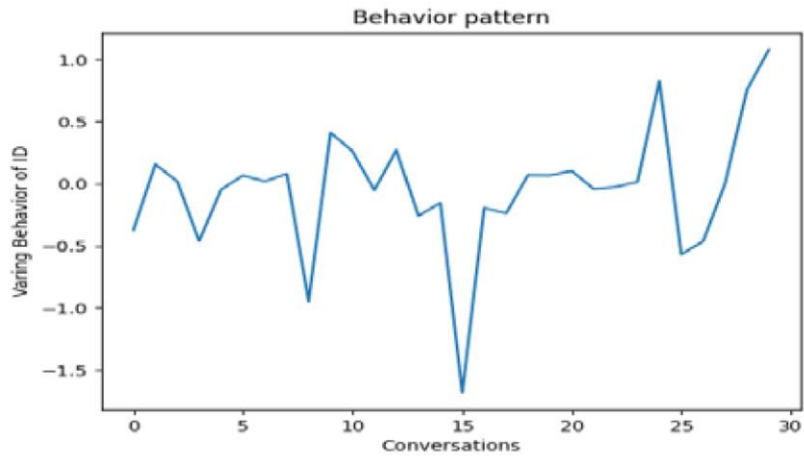


Figure 4(B). Behaviour pattern of ID₂

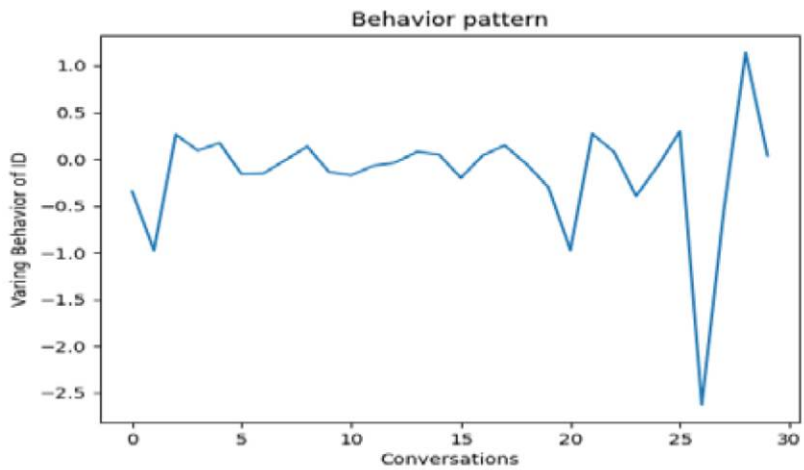
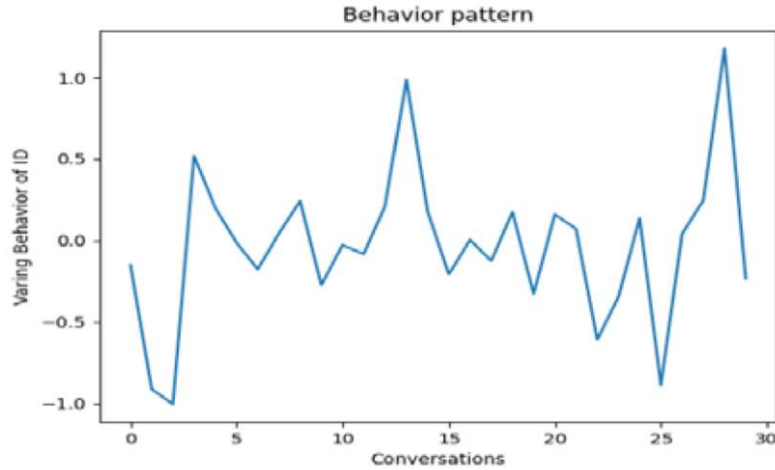


Figure 4(C). Behaviour pattern of ID₃

Figure 4(D). Behaviour pattern of ID₄

4.2 Real data based results

A total of 70% of the datasets were used for training, with the remaining 30% for testing the system. Results were evaluated for nine conversations among 100 IDs who participated in opinion sharing at different intervals of time. The proposed approach has been tested on two different datasets, one of which consists of simple sentences and the other of which is a complex dataset generated from speech (Table 6).

Table 6. Parameter of the datasets

Dataset1 (Simple dataset)				Dataset2 (Complex dataset)			
No of Conversations	(Min-Max) no of comments per conversation	(Min-Max). no of intervals per conversation		No of conversations	(Min-Max) no of comments per conversation	(Min-Max). no of intervals per conversation	
9	289	1609	23 25	9	1473	11481	18 23

4.2.1 Classification-based emotion, mood, and sentiment prediction

The employed technique is temporal information extraction, in which each word representing mood, sentiments, or emotions is assigned a weight. For each conversation, the features, including emotions, sentiments, and moods, are predicted for certain IDs using multinomial Naïve Bayes and logistic regression approaches for both datasets (Table 7 and Table 8). Experiments show that for simple or well-defined sentences, prediction accuracy is high, while for sentences with mixed emotions or very short sentences, prediction accuracy is low. However, prediction accuracy should be good because our model uses data from the prediction module to calculate behaviour.

Table 7. Comparison of classification techniques of emotions, moods, and sentiments for Dataset1

Emotions	Naive Bayes Approach			Logistic Regression Approach		
	Precision	Recall	F1 score	Precision	Recall	F1 score
Anger	0.95	0.09	0.17	0.83	0.72	0.77
Fear	0.94	0.06	0.12	0.84	0.62	0.71
Joy	0.57	0.98	0.72	0.88	0.75	0.81
Love	1.00	0.01	0.01	0.54	0.83	0.65
Sadness	0.64	0.88	0.74	0.78	0.88	0.83
Surprise	0.00	0.00	0.00	0.56	0.78	0.65
Mood						
Good	0.87	0.98	0.92	0.88	0.88	0.88
Bad	0.97	0.82	0.89	0.86	0.86	0.86
Sentiment						
Pos	0.87	0.98	0.92	0.88	0.88	0.88
Neg	0.97	0.82	0.89	0.86	0.86	0.86

Table 8. Comparison of classification techniques of emotions, moods and sentiments for Dataset 2

Emotions	Naive Bayes Approach			Logistic Regression Approach		
	Precision	Recall	F1 score	Precision	Recall	F1 score
Anger	0.06	0.00	0.00	0.19	0.26	0.22
Fear	0.00	0.00	0.00	0.02	0.01	0.01
Joy	0.32	0.06	0.10	0.31	0.18	0.23
Sadness	0.00	0.00	0.00	0.09	0.05	0.07
Surprise	0.34	0.05	0.08	0.24	0.17	0.20
Disgust	0.00	0.00	0.00	0.03	0.02	0.02
Mood						
Good	0.40	0.59	0.47	0.41	0.49	0.45
Bad	0.37	0.18	0.24	0.36	0.45	0.40
Sentiment						
Pos	0.39	0.11	0.17	0.39	0.20	0.26
Neg	0.39	0.71	0.51	0.41	0.52	0.46

4.2.2 Behaviour recognition applying proposed mathematical model

The conversation is considered as a series of comments. Therefore, m , s , and e are predicted for each comment, while event e_v and event intensity Ψ are calculated using the entire conversation. Finally, the result is obtained by putting all values in equation 5. For real datasets belief, cause and environment factor has been considered between 0 and 1. A tag or label has been assigned to participating IDs to maintain social trust on social networks (Table 9).

Table 9. Assigning social trust labels

Data	No of IDs	Polarity	ID with social trust label
Dataset 1	36	Negative	ID4, Most negative
	46	Positive	ID3, Most positive
	18	Neutral	Neutral
Dataset 2	58	Negative	ID19, Most negative
	33	Positive	ID82, Most positive
	9	Neutral	Neutral

The below-given graphs (Figure 5 and Figure 6) for both datasets show the most negative IDs, the most positive IDs, and those who remain neutral during all conversations. The results obtained from random observations show that an individual may have either positive or negative feelings or mixed feelings, which have been considered neutral and denoted by the PN state. A person turns positive after being neutral for a long time or would behave negatively under some discussion.

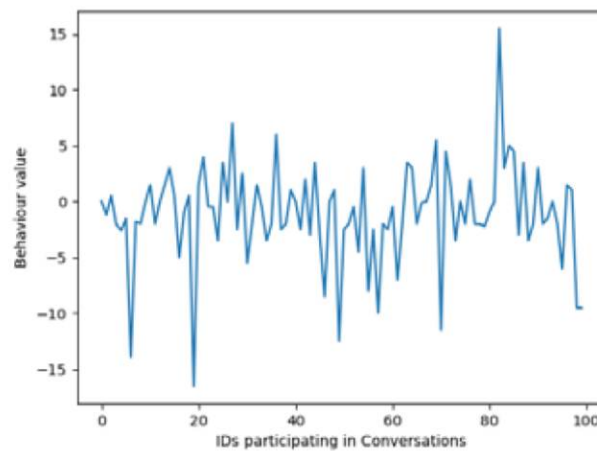


Figure 5. Behavior of each ID for Dataset 1

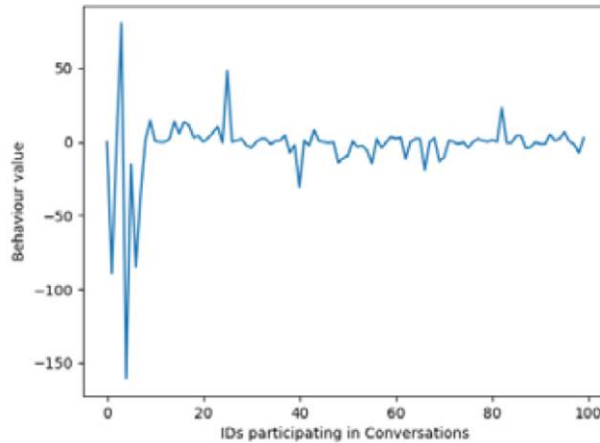


Figure 6. Behavior of each ID for Dataset 2

The following cases have been concluded after analyzing the obtained results:

Case 1: If a person stays in the P state, it indicates that the person is harmless.

Case 2: If a person stays in N state for various time-duration, he or she should be removed or warned, then removed.

Case 3: If a person changes states like $P \rightarrow P \rightarrow N \rightarrow P$; it's normal for an individual to get emotional at the point under discussion.

When we created the Trust State log for real data it depicted that some IDs remain in a positive state or neutral state for instance ID3(dataset 1) while ID4(dataset 1) mostly uses bitter wordings so filter-out as a negative person or might harmful for others.

5. Conclusion

In the presented research work, an ensemble model has been proposed that detects human behavior using a mathematical model combined with natural language processing from textual data. The presented study shows that human behavior can be modeled as the function of mood, sentiments, and feelings of some individuals. Fennell et al., employed structured sum-of-squares decomposition (S3D) for the data such as stock exchange data, to identify the human behavior. S3D claims that it gives good performance for classification. If we compare S3D and proposed mathematical behavior model conceptually, it shows that NLP and mathematical formulation give a blend with vivid clarity for the structured sentences. Our model also requires less number of features. Simulation-based results show that the model is effectively performing the tasks where we used random numbers. For real datasets, it has been shown that runtime prediction of emotions and sentiments has been done with good accuracy for simple sentences. However, for complex and very short sentences, prediction accuracy was low. As stated in Section 3.3.1, belief, emotion's cause, and environmental elements are assumed to have constant values equal to 0.5, which is a limitation of the proposed work. Behavior state estimation can help us to predict the positivity or

the negativity of some individuals that may ensure social trust on social networks. The proposed model evaluates and predicts the behavioral actions that can help achieve the following:

1. Textual data can be a good tool to reveal the hidden intentions, thoughts, sentiments, or emotions which can consequently help predict the forthcoming activities of individuals in term of positive or negative attitude. Any individual who remains in negative state, according to our model possesses negative nature and could be ignored.
2. A huge amount of data is available on social networks and, by applying machine learning techniques, hidden thoughts in terms of words might be discovered from texts.
3. Behavior at various times can assist in filtering out people with negative or positive intentions
4. Based on the mathematical modeling, a software system can give a suggestion to the individual about other user's trust level.
5. Behavior identification through technology can resolve the fear of scamming efficiently.

6. Future Work

In the future work, proposed technique will be applied to compute the behavior of social network users in the context of some specific events. Using evaluation of text in terms of mood, emotions, and sentiments, and machine learning; a categorization model for different actions with respect to some specific event will be designed.

Conflicts of interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

- 1 Kolliakou, A., Bakolis, I., Chandran, D., Derczynski, L., Werbeloff, N., Osborn, D. P., ... & Stewart, R. (2020). Mental health-related conversations on social media and crisis episodes: A time-series regression analysis. <https://doi.org/10.1038/s41598-020-60245-w>
- 2 Leary, M. (2012). Understanding the Mysteries of Human Behavior. <https://www.thegreatcourses.com/courses/understanding-the-mysteries-of-human-behavior>
- 3 Gutierrez, E., Karwowski, W., Fiok, K., Davahli, M. R., Liciaga, T., & Ahram, T. (2021). Analysis of human behavior by mining textual data: Current research topics and analytical techniques. <https://doi.org/10.3390/sym13071276>
- 4 Dinesen, P. T., Schaeffer, M., & Sønderkov, K. M. (2020). Ethnic diversity and social trust: A narrative and meta-analytical review. <https://doi.org/10.1146/annurev-polisci-051517-012706>
- 5 Kim, S. H., & Kim, S. (2021). Social trust as an individual characteristic or societal property? <https://doi.org/10.1080/12294659.2021.1874819>
- 6 Brass, D. J., Butterfield, K. D., & Skaggs, B. C. (1998). Relationships and unethical behavior: A social network perspective. <https://doi.org/10.5465/amr.1998.192955>
- 7 Ajzen, I., & Fishbein, M. (1977). Attitude-behavior relations: A theoretical analysis and review of empirical research. <https://doi.org/10.1037/0033-2909.84.5.888>

- 8 Sheeran, P., & Webb, T. L. (2016). The intention–behavior gap. <https://doi.org/10.1111/spc3.12265>
- 9 Fennell, P. G., Zuo, Z., & Lerman, K. (2019). Predicting and explaining behavioral data with structured feature space decomposition. <https://doi.org/10.1140/epjds/s13688-019-0191-x>
- 10 Torre, J. B., & Lieberman, M. D. (2018). Putting feelings into words: Affect labeling as implicit emotion regulation. <https://doi.org/10.1177/1754073917719332>
- 11 Mao, X., Chang, S., Shi, J., Li, F., & Shi, R. (2019). Sentiment-aware word embedding for emotion classification. <https://doi.org/10.3390/app9071334>
- 12 Rączy, K., & Orzechowski, J. (2021). When working memory is in a mood: Combined effects of induced affect and processing of emotional words. <https://doi.org/10.1007/s12144-019-00209-5>
- 13 Bennett, D., Davidson, G., & Niv, Y. (2022). A model of mood as integrated advantage. <https://doi.org/10.1037/rev0000337>
- 14 Riedl, M. O. (2019). Human-centered artificial intelligence and machine learning. <https://doi.org/10.1002/hbe2.117>
- 15 Kajić, I., Schröder, T., Stewart, T. C., & Thagard, P. (2019). The semantic pointer theory of emotion: Integrating physiology, appraisal, and construction. <https://doi.org/10.1016/j.cogsys.2018.10.003>
- 16 Alzhrani, K. M. (2022). Political Ideology Detection of News Articles Using Deep Neural Networks. <https://doi.org/10.32604/iasc.2022.022626>
- 17 Al-Samadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2018). Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' review. <https://doi.org/10.1016/j.jocs.2018.05.019>
- 18 Plonsky, O., Apel, R., Ert, E., Tennenholtz, M., Bourgin, D., Peterson, J. C., ... & Erev, I. (2019). Predicting human decisions with behavioral theories and machine learning. <https://arxiv.org/abs/1904.06866>
- 19 Yu, Z., Du, H., Yi, F., Wang, Z., & Guo, B. (2019). Ten scientific problems in human behavior understanding. <https://doi.org/10.1007/s42486-019-00007-4>
- 20 Emanuel, A., & Eldar, E. (2023). Emotions as computations. <https://doi.org/10.1016/j.neubiorev.2023.104977>
- 21 Deng, S., Xia, S., Hu, J., Li, H., & Liu, Y. (2021). Exploring the topic structure and evolution of associations in information behavior research through co-word analysis. <https://doi.org/10.1177/0961000620917711>
- 22 Gross, J. J. (1998). The emerging field of emotion regulation: An integrative review. <https://doi.org/10.1037/1089-2680.2.3.271>

- 23 Satu, M. S., Khan, M. I., Mahmud, M., Uddin, S., Summers, M. A., Quinn, J. M., & Moni, M. A. (2021). TClustVID: A novel machine learning classification model to investigate topics and sentiment in COVID-19 tweets. <https://doi.org/10.1016/j.knosys.2021.107126>
- 24 Almeida, A., & Azkune, G. (2018). Predicting human behavior with recurrent neural networks. <https://doi.org/10.3390/app8020305>
- 25 Pentland, A., & Liu, A. (1999). Modeling and prediction of human behavior. <https://doi.org/10.1162/089976699300016890>
- 26 Balaji, B. S., Balakrishnan, S., Venkatachalam, K., & Jeyakrishnan, V. (2021). Automated query classification based web service similarity technique using machine learning. <https://doi.org/10.1007/s12652-020-02591-5>
- 27 Machová, K., Szabóová, M., Paralič, J., & Mičko, J. (2023). Detection of emotion by text analysis using machine learning. <https://doi.org/10.3389/fpsyg.2023.1190326>
- 28 Russell, J. A. (1980). A circumplex model of affect. <https://doi.org/10.1037/h0077714>

Predicting and Characterizing piRNAs and their Functions Using an Integrated Machine Learning Approach

Anam Umera^a, Sajid Mahmood^a, Usman Inayat^{a*}

^a School of Systems and Technology, University of Management and Technology
anamumera333@gmail.com, sajid.mahmood@umt.edu.pk, usman.inayat@umt.edu.pk

*Corresponding Author: Usman Inayat usman.inayat@umt.edu.pk

Abstract

One class of short non-coding RNA molecule that is well recognized is called PIWI-interacting RNA (piRNA). PiRNAs are involved in the creation of novel medications as well as the identification of different kinds of tumors. Additionally, it is associated with stopping transposes, managing gene transcription, and maintaining genomic integrity. The important role that piRNAs play in biological processes has led to a growing body of research in bioinformatics on the discovery of piRNAs and their functionality. In this research, a powerful model is proposed to improve PiRNA prediction and functionality. The suggested model uses four classifiers (Logistic Regression, SVC, Random Forest, and Gradient Boosting Classifier) for classification. Moreover, TNC and DNC are used to acquire features. There are two layers involved in developing the suggested model. A sequence's potential to be piRNA is predicted in the first layer, and its potential to direct target mRNA deadenylation is predicted in the second. In the first layer, the model's accuracy is 98.59%, and in the second layer, it is 94.55%.

Keywords: PIWI-interacting RNAs, Intelligent Model for PiRNAs, Prediction of PiRNAs, Machine learning based model, Functions of PIWI-interacting RNAs.

List of Abbreviations:

RNA	:	RiboNucleic Acid
PiRNA	:	Piwi-interacting RNAs
RRNAs	:	Ribosomal RNAs
SnoRNAs	:	Small nucleolar RNAs
ncRNA	:	Non-coding RNA
SiRNAs	:	Small interference RNAs
TRNAs	:	Transfer RNAs
MRNA	:	Messenger RNA
LncRNA	:	Long non-coding RNAs
BC	:	Breast Cancer
ORFs	:	Open Reading Frames
OH	:	Hydroxide
PIWI	:	P-element Induced Wimpy
PiRISC	:	PiRNA-Induced Silencing Complex

DNA	:	Deoxyribonucleic Acid
DNMT	:	DNA Methyl Transferase
ShRNA	:	Short hairpin RNA
GC	:	Gastric Cancer
HCC	:	Hepatocellular Carcinoma
TEs	:	Transposable Elements
ISCs	:	Intestine Stems Cells
CLIP	:	Cross-Linking and ImmunoPrecipitation
LFE-GM	:	Luca Fuzzy Entropy and Gaussian Membership
SRA	:	Sparse Recognition Algorithm
FCS	:	Feature Score Criteria
DAC	:	Di-Nucleotide Auto Covariance
DNN	:	Deep Neural Network
GB	:	Gradient boosting
SVM	:	Support Vector Machine
RF	:	Random Forest
LR	:	Logistic Regression
DNC	:	Di-Nucleotide Composition
TNC	:	Tri-Nucleotide Composition

1. Introduction

Through a variety of adjustments, dynamic balance in human health is achieved. Normal daily routines require the internal environment to be in a stable state. Internal environment disturbances lead to mutations and imbalances in genes, which leads to a variety of human illnesses, often to the extent of cancer, heart and brain disorders, claiming millions of lives annually. The regular functional activities of the cells are primarily driven by several biochemical signaling systems and genetic modifications, maintaining the homeostasis of the internal environment [1]. PiRNA is small non-coding RNA molecule, having significant importance in the development of novel medications as well as the identification of different kinds of tumors, heart and brain disorders. Additionally, it is associated with stopping transposes', managing gene transcription, and maintaining genomic integrity [2].

Almost all non-coding RNAs have drawn a great interest for their involvement in cellular processes, and in detection of multiple disorders. All the regulating non-coding RNAs can be roughly split into smaller and large non-coding RNAs depending on the size of the molecule [3]. In prokaryotic cells, PIWI interacting RNA is a documented class of short non-coding RNA molecules having a polymer that is 24 to 31 nucleotides long. PiRNAs carry out a wide range of genetic and biological tasks, such as, controlling the activation of genes, preserving, and forming genetic material, and synthesis of specific protein [4].

2. Background

One class of short non-coding RNA molecule that is well recognized is called PIWI-interacting RNA (piRNA), distinct from other types, including such as microRNAs and siRNAs, which play crucial roles in the control of gene production in animal cells. Initially in early 2000s, piRNAs are predominantly found in the germ line cells of animals, although their presence and functions in

somatic cells have also been recognized [5]. These PiRNA molecules association with PIWI proteins, (a subclade of the Argonaut proteins), forms piRNA complexes, this interaction plays a fundamental role in silencing transposable elements, thereby preventing genomic instability and potential mutagenesis. The primary piRNAs role is to safeguard the genome's stability, achieving this by targeting and regulating transposable elements, which are sequences that can change their positions within the genome, often resulting in mutations and alterations in the cell's genetic material. By the formation of complexes with PIWI proteins, piRNAs facilitate the cleavage of transposon mRNA, thus preventing potentially harmful elements from being transcribed and interfering with normal cellular processes [6].

Additionally, piRNAs play a role in the epigenetic regulation of genes. They influence the modification of chromatin structure, thus aiding in the transcriptional silencing of genes. This regulatory capacity extends beyond transposons to include other genomic elements and is crucial during development, especially in germ cells where it ensures the transmission of genetic information across generations without disruptions from transposable elements. The study of piRNAs holds significant clinical implications, particularly in the field of oncology. Aberrant expression of piRNAs has been associated with various types of cancers, implying that they might act as biomarkers for the detection and prediction of cancer. Moreover, because piRNAs contribute to the maintenance of genomic stability, understanding their mechanisms can provide insights into the progression of cancer, where genomic instability is a hallmark [7]. The potential role of piRNAs in therapeutics is also being explored. As regulators of gene expression, targeting piRNA pathways, could offer new avenues for the treatment of diseases that arise from genetic and epigenetic dysregulations, such as cancer and hereditary disorders. Modulating piRNA activity might help to correct or compensate for pathological gene expression profiles. Recent studies indicate that piRNAs are widely transcribed in a variety of physiological cell types and are associated with a wide range of clinical conditions besides those that have been documented in the germ cells. PiRNAs, for instance, have been found to explain inordinately in many diseases [8].

PiRNAs have also demonstrated promising results as prognostic indicators for a variety of tumors. The dysregulation of piRNAs in cancer suggests that they may be potential targets in cancer therapy. The most prevalent malignancy and the leading reason for tumor-related deaths in women is breast tumor. PiRNAs influence a significant impact in breast tumor and may be used as diagnostics and treatment approaches. The excessive transcription of piRNAs is seen in tumors and is linked to proliferation of the cancer cells. Additionally, there is an indication that piRNA-mediated genomic mechanisms contribute to cancer. In breast tumor cells, piR-651 was shown to be substantially amplified. One of the biggest reasons of cancer-related deaths worldwide is lung cancer. A possible diagnostic and treatment strategy in lung cancer involves excessive piRNA transcription, therefore targeting it might be a possible tool in limiting the cancer spread and growth. In laboratory and in mammalian lung cancer, progression might be inhibited by piR-55490, which was also inversely correlated with patients' survival. While other research has established that in lung cancer cells piR-651 is enhanced, piR-L-163 was the piRNA that was most frequently down-regulated in lung cell lung cancer when opposed to the equivalent non-tumor lung cells [9].

3. Literature Review

For recognizing sequences both piRNA and non-piRNA, numerous models have been proposed that are computerized. To predict piRNA, a sequential computational model called "piRNA-CNN" based on convolutional neural network is proposed [11]. DNC and TNC are used for feature extraction. In [12] presented 2L-piRNA, an effective predictor. It's an ensemble model with two layers.-The 1st layer used to determine whether a dataset is a PiRNA or non-PiRNA, and the 2nd layer used to determine whether a piRNA can direct the de-adenylation of a target mRNA or not. For classification, SVM classifier, CNN models are used and pseudo dinucleotide composition ($k=2$) for feature extraction. "2S-piRCNN", a two-stage deep-learning classifier that makes use of a CNN, is proposed in [13]. In [14] "2L-piRNAPred" model is proposed. 2L-piRNAPred is SVM-based predictor. 2-layer merged Scheme for point out piRNAs in the 1st layer and identifying if piRNA have the task of directing selected mRNA de-adenylation in the 2nd layer. In [17] proposed "piRNN" for identifying piRNA. Convolutional neural network classifiers were used, each of which had been trained using datasets from four organisms. Each sequence was represented by a matrix of k -mer frequencies. In [18] "2L-piRNADNN" model was proposed. To reduce computing complexities via parallel processing, it is suggested to use the DNN model with the Spark computing platform. A feature vector consisting of numerical values that was created from the RNA sequences by the suggested model's use of the dinucleotide auto covariance approach.

In [19] created a quick, reliable, and effective deep learning technique called piRDA for locating the correlations between piRNAs and diseases. Without using any features of engineering, the suggested architecture takes the most important and information that is generically expressed in a piRNA disease pair from the unprocessed sequences. K fold cross validation is used to assess the effectiveness of the suggested method piRDA. In contrast to community methods, the piRDA greatly outperforms them all in terms of quality assessment criteria for the detection of piRNA disease connections. In [20], they provide an integrated strategy for piRNA prediction that considers a range of genetic and epigenetic traits that can be utilized to describe these molecules. They have gathered and examined a sizable variety of piRNA characteristics that have been empirically verified in numerous species. In an object-oriented framework that uses a Various Kernel Learning technique, these properties are expressed by several kernels. The developed tool, known as IpiRIId, outperforms all other tools with prediction findings that reach more than 90% accuracy for the three examined species. Additionally, their method enables researchers to examine the applicability of each specified trait in a particular specie.

The proposed method can also be modified to anticipate different types of ncRNAs because it is modular and easily expandable. IpiRIId model performance in term of Acc, Sp, Sn is 93.66%, 96.58%, 90.74% respectively. For classification, SVM classifier, CNN models are used and pseudo dinucleotide composition ($k=2$) for feature extraction. "2S-piRCNN", a two stage deep learning classifier that makes use of a CNN, is proposed in [21]. In this [22] study, they created the 2lpiRNAPred a combined algorithm with two layers' approach. In the first layer it recognizes piRNAs and assesses if they are involved in the second layer process of inducing target mRNA deadenylation. To make the attributes' dimensions smaller, a new feature extraction approach depends on Gaussian and Luca fuzzy entropy participation function (LFE-GM) was presented. Two types of classifiers—Sparse Recognition Algorithm (SRA) and Five attribute detection techniques using the Support Vector Machine with Mahalanobis Proximity Dependent Rotational

Basis Process: Extended serial similarity pseudo dinucleotide composition, k-mer, overall sequence similarity pseudo dinucleotide composition, Standardized Moreau Broto auto-correlation, and Geary auto-correlation—were combined to create the unified classifier method with two layers. The outcomes show that 2lpiRNAPred outperforms six other available prediction methods by a wide margin. 2lpiRNAPred model performance in term of Acc, Sp, Sn, MCC is 88.72%, 85.54%, 91.89%, 0.775 respectively.

To improve the use of deep learning methods for piRNA and function identification, a two-layer predictor is proposed in this [23] study. The suggested approach uses multiple feature acquisition methods to consider the physical and chemical characteristics of the biological sequences while extracting features. The k-fold cross-validation approach is used to thoroughly assess the suggested strategy output. The evaluation's findings indicate that the suggested model outperformed the current models, with accuracy gains of 7.59 and 2.81 percent at layer 1st and layer 2nd, correspondingly. The suggested paradigm is believed to be useful for precision medicine and cancer diagnostics. In [24], they address current developments in our knowledge of piRNA function as well as prospective testing and treatment uses of piRNAs in a variety of digestion malignancies. In order to predict connections between piRNAs and disorders using information-retrieving technology, they present a unique predictor dubbed iPiDA-LTR. According to research findings, iPiDA-LTR shows promising results in detecting disorders linked to both previously identified and newly discovered piRNAs.

In this study, a powerful model is proposed to improve PiRNA prediction and its functionality. The proposed model uses four classifiers (Logistic Regression, SVC, Random Forest, and Gradient Boosting Classifier) for classification. Moreover, TNC and DNC are used to acquire features. There are two layers involved in developing the suggested model. A sequence's potential to be piRNA is predicted in the first layer, and its potential to direct target mRNA deadenylation is predicted in the second.

This contribution is organized as follows: Section 4 represent material and methods. Section 5 represents the results and discussion of the suggested model. Finally, in Section 6, a conclusion and future directions are offered.

4. Materials and methods

The following five step are used to complete the proposed model as shown in Figure 1: (1) data acquisition and analysis, (2) feature extraction (DNC, TNC), (3) training the three ML algorithms, (4) generating new features on the basis of the three ML model's outputs of step # 3 and the original dataset, and (5) training the one ML classifier. The detail of these stages is given below.

Same model and stages are used for both “Dataset 1” and “Dataset 2” prediction. As Shown in section 4.1 “Dataset 1” represent S and “Dataset 2” represent S+.

4.1. Data Acquisition and Analysis

The same dataset as in [20] is used. There are 709 piRNA sequences that instruct target mRNA deadenylation (denoted as S_{inst}^+), 709 piRNA sequences that do not have this function (denoted as $S_{non-inst}^+$), and 1418 non-piRNA sequences. PiRNA sequences denoted as S^+ and non-PiRNA sequences denoted as S^- . Consequently, the datasets for this investigation can be described as follows:

$$S = S^+ \cup S^- \dots \dots (1)$$

$$S^+ = PiRNA \dots \dots \dots (2)$$

$$S^- = NonPiRNA \dots \dots (3)$$

$$S^+ = S_{inst}^+ \cup S_{non-inst}^+ (4)$$

S represents “Dataset 1” and S+ represents “Dataset 2”.



Figure 1. 5 Steps of the Proposed model

4.2. Feature Extraction

The piRNA and non-PiRNA sequence dataset is used to extract features, and two feature encoding techniques are used (i.e., DNC, TNC). In this research, iLearn web server is used for feature extraction. DNC is used to extract 16 features and TNC to extract 64 features in total.

a. Di-Nucleotide Composition (DNC)

Feature extraction approach that uses a set of 2 sequential nucleotides to represent an RNA sequence is called Di-Nucleotide Composition (DNC). The probability of each set, denoted by N1 N2 for the first set, N2 N3 for the second set, and so on, is calculated [12]. 16 features are provided by the di-nucleotide composition. It's described as:

$$D(a, b) = \frac{Nab}{n-1}, a, b \in \{A, C, G, (U)\} \dots\dots(5)$$

In equation (5) Nab is the number of Di-Nucleotide described by nucleic acid types a and b. For example, ab ∈ {AA, AC, . . . L, GT, TT} sequences.

b. Tri-Nucleotide Composition (TNC)

Another feature extraction approach is called Tri-Nucleotide Composition (TNC), which uses three pairs of sequential nucleotides to describe an RNA sequence. Each pair's probability is computed. For instance, the first set in an RNA sequence is N1 N2 N3, the second set is N2 N3 N4, and so on, producing a $4 \times 4 \times 4 = 64D$ matching features vector [10]. It's outlined as:

$$D(a, b, c) = \frac{N(abc)}{n-2}, a, b, c \in \{A, C, G, (U)\} \dots\dots(6)$$

In equation (6) $Nabc$ is the number of Tri-Nucleotide described by nucleic acid types a, b and c . For example, $abc \in \{AAA, AAC, AAG, \dots, TTT\}$ sequences.

4.3. Train Machine Learning Classifiers

We merge DNC and TNS features and make a dataset. Before experiment, we check the sub features contribution in overall dataset. So, we select best top 10 sub features from dataset using Random Forest Regressor. Figure 2 presents the important sub features. All sub features range is 0.00 to 0.05, but CG sub feature range is 0.35. So, we normalize the CG sub feature with 0.15.

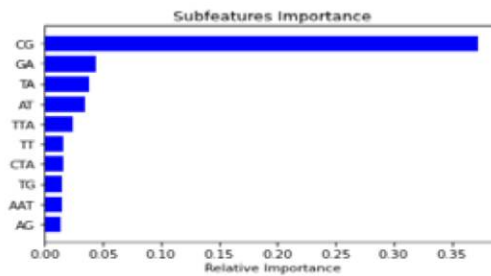


Figure 2. Top 10 important Sub-features with their relative importance

Figure 3 represents a workflow for creating a machine learning model pipeline using a combination of classifiers and a meta-classifier. Figure 4 represents the same workflow as Figure 3, but the difference between Figure 3 and 4 is that Figure 3 is used for "Dataset 1" and Figure 4 is used for "Dataset 2". "Dataset 1" as shown in session 2.1 contain PiRNA and non-PiRNA data sequences. And "Dataset 2" contains samples having the function of instructing target RNA deadenylation and samples without this function.

The process begins with loading the initial dataset, referred to as "Dataset 1" and "Dataset 2". Three classifiers (Logistic Regression, SVM, and Random Forest) are used for prediction. The predictions from the three classifiers are combined with the original features from the test set to create a new dataset. The new dataset, which includes both the original features and the predictions from the three classifiers, is used to train a meta-classifier. In this case, the meta-classifier is a Gradient Boosting Classifier. The Gradient Boosting Classifier is trained on the new dataset, and its accuracy is evaluated to assess the achievement of the overall model. The proposed model has been developed in Python using Google Colab. We employed the Holdout Method to divide the dataset into two parts: 80% for training and 20% for testing.

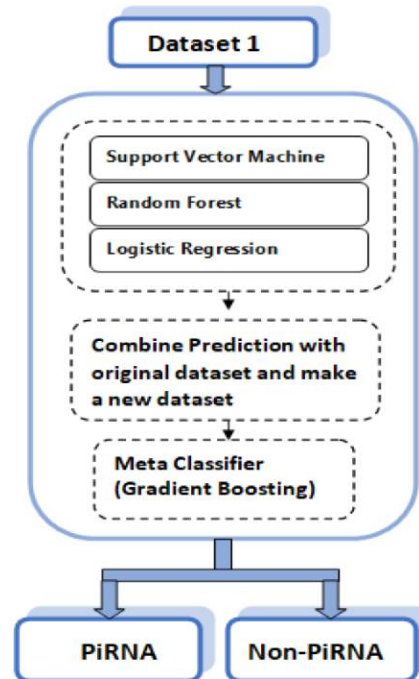


Figure 3. Proposed Model for Dataset 1

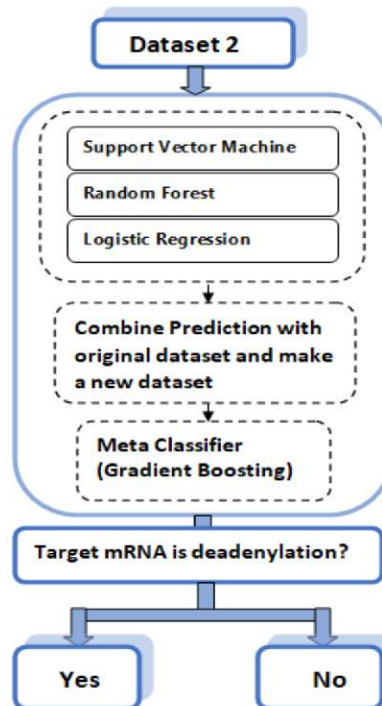


Figure 4. Proposed Model for Dataset 2

5. Results & Discussion

In this section, the experimental results of the suggested model for both Dataset 1 and Dataset 2 are discussed. As Figure 3 indicates, when Dataset 1 is applied to the suggested model, the results are obtained in the form of PiRNA or Non-PiRNA. Whereas, when Dataset 2 is applied, the results were obtained in the form of target mRNA deadenylation or no deadenylation.

5.1. Results of the Proposed Model for Dataset 1

Figure 5 provides a visual summary of the model's performance across various metrics: Precision, AUC (Area Under the Curve), MCC (Matthews Correlation Coefficient), F1-Score, Balanced Accuracy, Specificity, Sensitivity (Recall), and Accuracy.

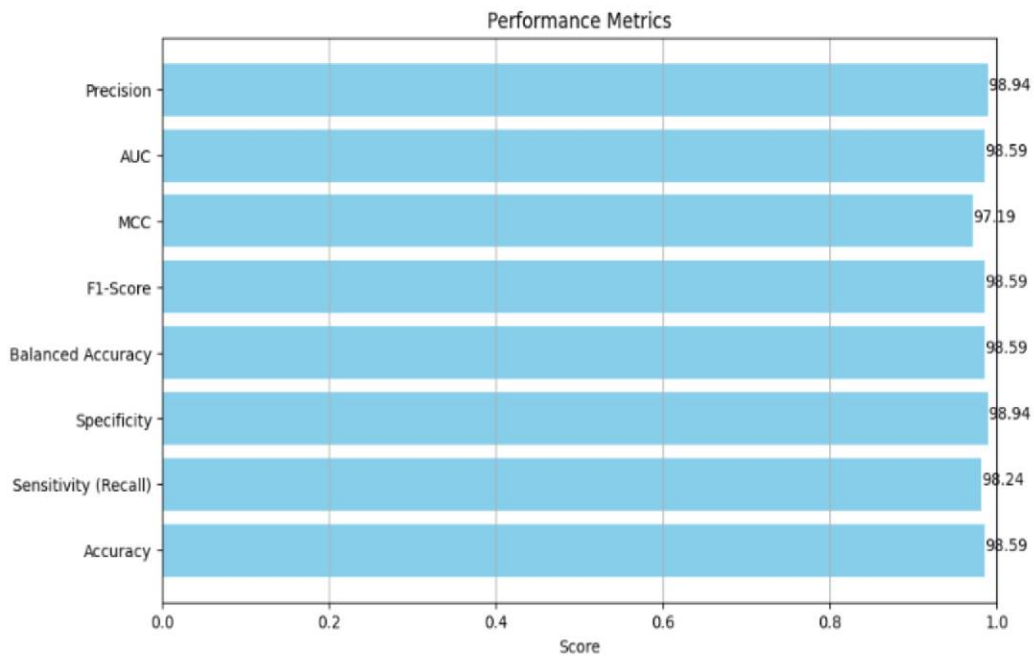


Figure 5. Results of the Proposed Model for Dataset 1

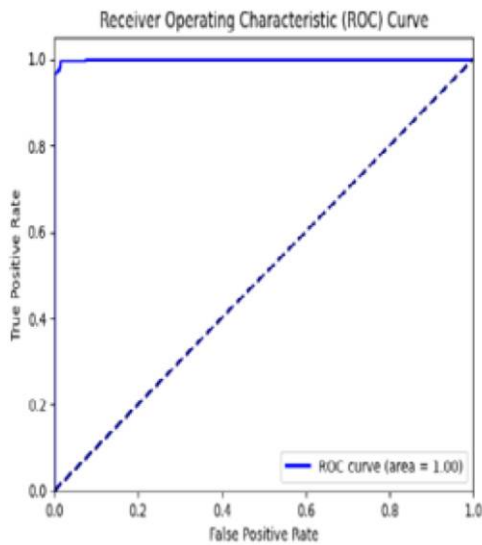


Figure 6. ROC Curve for Dataset 1

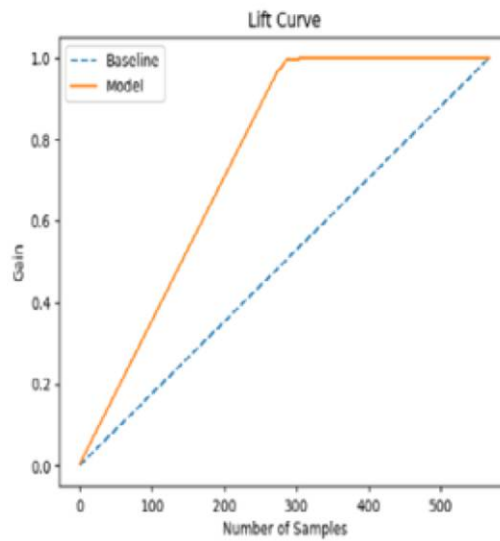


Figure 7. Lift Curve for Dataset 1

Figure 6 shows, a Receiver Operating Characteristic (ROC) Curve, which plots the True Positive Rate (TPR, also known as Sensitivity or Recall) against the False Positive Rate (FPR) for different threshold settings of a binary classification model. In the figure.6, performance of the model shows using blue line. In this case: the ROC curve is almost touching the top-left corner, indicating excellent model performance. The Area Under the Curve (AUC) is 1.00, suggesting a perfect classification ability of the proposed model. The lift curve, depicted in Figure. 7, compares the outcomes produced with and without a predictive model to determine how effective the model is. The performance of a random model is represented by the horizontal line, which is the baseline. The lift, or the ratio of the model-assisted outcomes to the model-off outcomes, is represented by the vertical axis.

5.2. Results of the Proposed Model for Dataset 2

Figure 8 shows, a visual summary of the model's achievement for Dataset 2 across various metrics. These metrics collectively deliver a comprehensive overview of the model's achievement, ensuring it is robust across different aspects of prediction quality. Figure 8 illustrates these metrics, emphasizing the model's capability to effectively classify the data in Dataset 2, which involves distinguishing between target mRNA deadenylation and not.

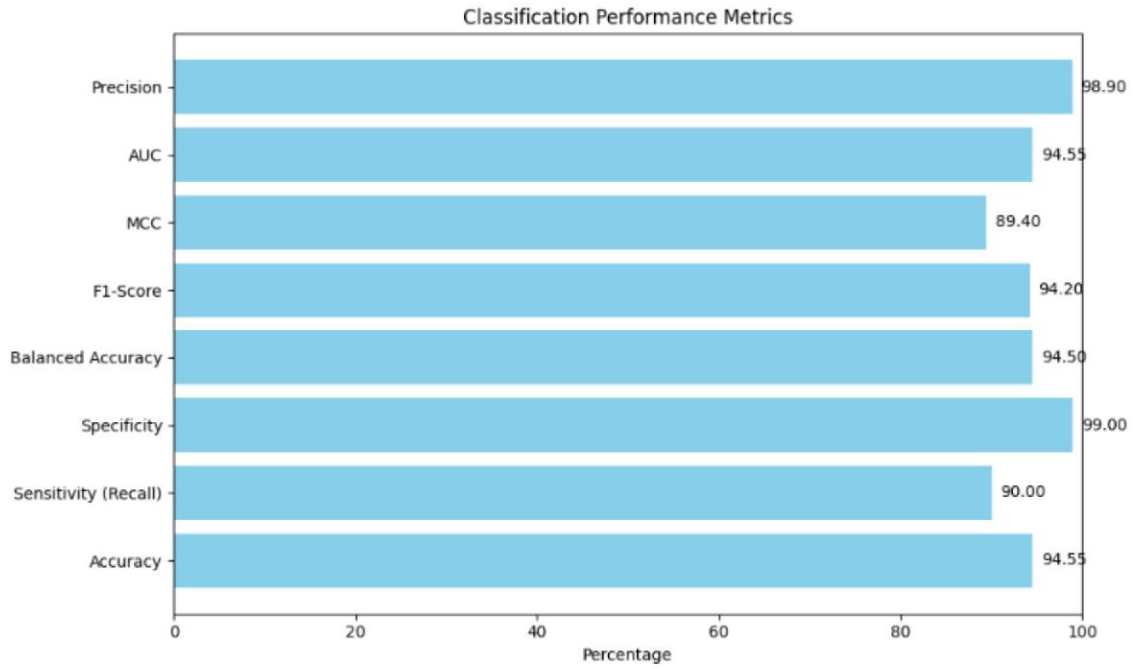


Figure 8. Results of the Proposed Model for Dataset 2

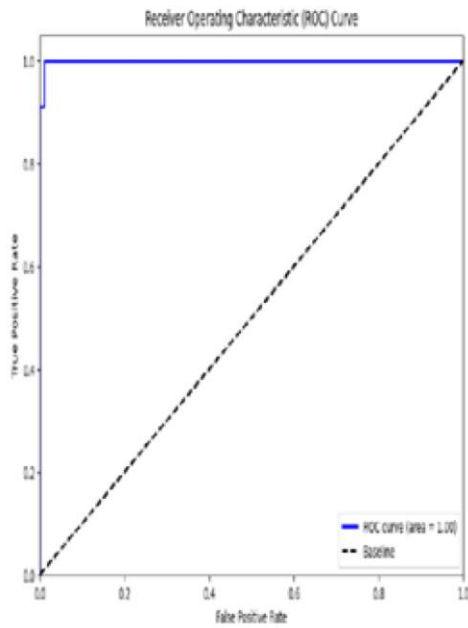


Figure 9. ROC Curve for Dataset 2

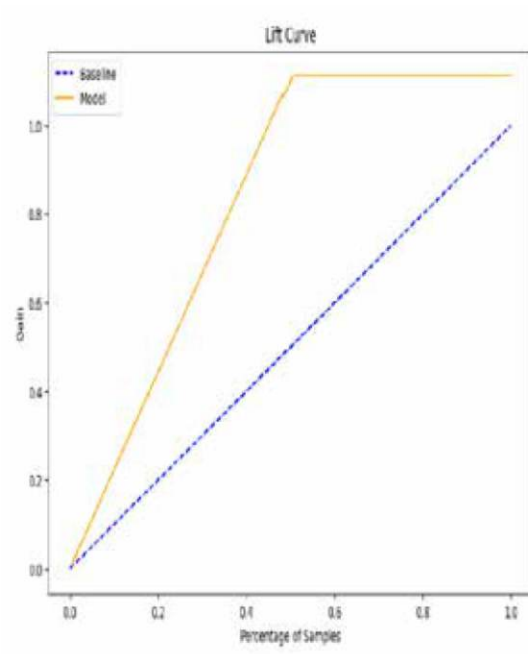


Figure 10. Lift Curve for Dataset 2

Figure 9 shows, a Receiver Operating Characteristic (ROC) Curve for Dataset 2. Similarly, figure 10 shows, Lift Curve for Dataset 2.

The proposed model demonstrates significant efficacy in classifying piRNA and non-piRNA sequences, as well as in distinguishing piRNA samples based on their function in target mRNA deadenylation. The high accuracy and other performance metrics obtained for both Dataset 1 and Dataset 2 indicate that the combined feature extraction techniques (DNC and TNC) and the machine learning approach are effective in capturing the relevant features and patterns within the data. To check the model performance, we use various metrics: Accuracy, Balance Accuracy, Precision, AUC (Area Under the Curve), MCC (Matthews Correlation Coefficient), F1-Score, Balanced Accuracy, Specificity, and Sensitivity (Recall).

For Dataset 1, which involves the classification of piRNA and non-piRNA sequences, the success rate of the suggested model is expressed in terms of Acc, Sn, Sp, BalanceAcc, F1-Score, MCC, AUC, and Precision are 98.59%, 98.24%, 98.94%, 98.59%, 98.59%, 97.19%, 98.59%, 98.94% respectively. The results suggest that the proposed feature extraction techniques and model architecture can effectively differentiate between piRNA and non-piRNA sequences.

For Dataset 2, which classifies piRNA samples based on their ability to instruct, target mRNA deadenylation. For Dataset 2, the suggested model success rate is expressed in the form of ACC, Sn, Sp, BalanceAcc, F1-Score, MCC, AUC, and Precision are 94.55%, 90.00%, 90.00%, 94.50%, 94.20%, 89.40%, 94.55%, 98.90% respectively.

Table 1. The suggested model is compared to the benchmark model [20].

Layers	Models	Acc (%)	Sp (%)	Sn (%)	Mcc (%)
1 st Layer	Proposed Model	99.59	98.94	98.24	97.19
	2L-piRNAPred	89	87.5	90.4	0.779
2 nd Layer	Proposed Model	94.55	90.00	90.00	89.40
	2L-piRNAPred	84.0	83.6	84.3	0.680

While the proposed model achieves excellent performance, several limitations should be acknowledged. First, the model's training and testing were conducted on a specific dataset, this could restrict the applicability of the findings to other piRNA datasets or different organisms. Second, the feature extraction methods used, while effective, are computationally intensive. This may pose challenges for scaling the model to very large datasets or real-time applications. Third, the current model focuses solely on sequence-based features. Incorporating additional types of data, such as secondary structure information or interaction networks, could potentially improve the model's performance.

6. Conclusions

This study introduces a robust and innovative model, aimed at enhancing the prediction and functional analysis of PIWI-interacting RNA (piRNA) sequences. By employing a combination of four advanced classifiers (Logistic Regression, Support Vector Classifier (SVC), Random Forest, and Gradient Boosting Classifier) the model effectively discriminates between piRNA and non-piRNA sequences, achieving a remarkable accuracy of 98.59% in this primary classification task.

Additionally, by utilizing features extracted through Dinucleotide Composition (DNC) and Trinucleotide Composition (TNC), the model further predicts the functional role of identified piRNA sequences, specifically their ability to instruct target mRNA deadenylation, with an accuracy of 94.55%. The high accuracy rates at both classification steps highlight the potential of the proposed model to contribute meaningfully to bioinformatics research and medical applications involving piRNAs.

Future research should aim to address the limitations mentioned in section 4 by validating the model on a wider range of datasets and exploring more efficient feature extraction techniques. Additionally, integrating multi-omics data, such as transcriptomics and proteomics, could enhance the understanding of piRNA function and their regulatory mechanisms.

References

1. Shoji, K., & Tomari, Y. (2024). A potential role of inefficient and non-specific piRNA production from the whole transcriptome. <https://doi.org/10.1101/2024.01.24.577019>.
2. Taverna, S., Masucci, A., & Cammarata, G. (2023). PIWI-RNAs Small Noncoding RNAs with Smart Functions: Potential Theranostic Applications in Cancer. *Cancers*, 15(15), 3912. <https://doi.org/10.3390/cancers15153912>.
3. Allikka Parambil, S., Li, D., Zelko, M., Poulet, A., & van Wolfswinkel, J. C. (2024). piRNA generation is associated with the pioneer round of translation in stem cells. *Nucleic Acids Research*, 52(5), 2590–2608. <https://doi.org/10.1093/nar/gkad1212>.
4. Huang, X., Wang, C., Zhang, T., Li, R., Chen, L., Leung, K. L., Lakso, M., Zhou, Q., Zhang, H., & Wong, G. (2023). PIWI-interacting RNA expression regulates pathogenesis in a *Caenorhabditis elegans* model of Lewy body disease. *Nature Communications*, 14(1), 6137. <https://doi.org/10.1038/s41467-023-41881-8>.
5. Liu, L., Li, L., Zu, W., Jing, J., Liu, G., Sun, T., & Xie, Q. (2023). PIWI-interacting RNA-17458 is oncofetal and a potential therapeutic target in cervical cancer. *Journal of Cancer*, 14(9), 1648–1659. <https://doi.org/10.7150/jca.83446>.
6. Wang, J., Zhu, S., Meng, N., He, Y., Lu, R., & Yan, G.-R. (2019). ncRNA-Encoded Peptides or Proteins and Cancer. *Molecular Therapy*, 27(10), 1718–1725. <https://doi.org/10.1016/j.ymthe.2019.09.001>.
7. López-Jiménez, E., & Andrés-León, E. (2021). The Implications of ncRNAs in the Development of Human Diseases. *Non-Coding RNA*, 7(1), 17. <https://doi.org/10.3390/ncrna7010017>.
8. Wang, K., Wang, T., Gao, X., Chen, X., Wang, F., & Zhou, L. (2021). Emerging functions of piwi-interacting RNAs in diseases. *Journal of Cellular and Molecular Medicine*, 25(11), 4893–4901. <https://doi.org/10.1111/jcmm.16466>.
9. Weng, W., Li, H., & Goel, A. (2019). Piwi-interacting RNAs (piRNAs) and cancer: Emerging biological concepts and potential clinical implications. *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer*, 1871(1), 160–169. <https://doi.org/10.1016/j.bbcan.2019.09.001>.
10. Yu, Y., Xiao, J., & Hann, S. S. (2019). The emerging roles of PIWI-interacting RNA in human cancers. *Cancer Management and Research*, Volume 11, 5895–5909.

<https://doi.org/10.2147/CMAR.S209300>.

11. Fathizadeh, H., & Asemi, Z. (2019). Epigenetic roles of PIWI proteins and piRNAs in lung cancer. *Cell & Bioscience*, 9(1), 102. <https://doi.org/10.1186/s13578-019-0368-x>.
12. Smyth, E. C., Nilsson, M., Grabsch, H. I., van Grieken, N. C., & Lordick, F. (2020). Gastric cancer. *The Lancet*, 396(10251), 635–648. [https://doi.org/10.1016/S0140-6736\(20\)31288-5](https://doi.org/10.1016/S0140-6736(20)31288-5).
13. Feng, J. X., & Riddle, N. C. (2020). Epigenetics and genome stability. *Mammalian Genome*, 31(5–6), 181–195. <https://doi.org/10.1007/s00335-020-09836-2>.
14. M. Tahir, M. Hayat, S. Khan, and K. to Chong, “Prediction of Piwi-Interacting RNAs and Their Functions via Convolutional Neural Network,” *IEEE Access*, vol. 9, pp. 54233–54240, 2021, doi: 10.1109/ACCESS.2021.3070083.
15. S. D. Ali, W. Alam, H. Tayara, and K. Chong, “Identification of Functional piRNAs Using a Convolutional Neural Network,” *IEEE/ACM Trans. Comput. Biol. Bioinform.*, pp. 1–1, 2020, doi: 10.1109/TCBB.2020.3034313.
16. T. Li, M. Gao, R. Song, Q. Yin, and Y. Chen, “Support Vector Machine Classifier for Accurate Identification of piRNA,” *Appl. Sci.*, vol. 8, no. 11, p. 2204, Nov. 2018, doi: 10.3390/app8112204.
17. Khan, S., Khan, M., Iqbal, N., Hussain, T., Khan, S. A., & Chou, K.-C. (2020). A Two-Level Computation Model Based on Deep Learning Algorithm for Identification of piRNA and Their Functions via Chou’s 5-Steps Rule. *International Journal of Peptide Research and Therapeutics*, 26(2), 795–809. <https://doi.org/10.1007/s10989-019-09887-3>.
18. Khan, S., Khan, M., Iqbal, N., Li, M., & Khan, D. M. (2020). Spark-Based Parallel Deep Neural Network Model for Classification of Large Scale RNAs into piRNAs and Non-piRNAs. *IEEE Access*, 8, 136978–136991. <https://doi.org/10.1109/ACCESS.2020.301150>.
19. Ali, S. D., Tayara, H., & Chong, K. T. (2022). Identification of piRNA disease associations using deep learning. *Computational and Structural Biotechnology Journal*, 20, 1208–1217. <https://doi.org/10.1016/j.csbj.2022.02.026>.
20. Ali, S. D., Alam, W., Tayara, H., & Chong, K. (2020). Identification of Functional piRNAs Using a Convolutional Neural Network. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1–1. <https://doi.org/10.1109/TCBB.2020.3034313>.
21. Zuo, Y., Zou, Q., Lin, J., Jiang, M., & Liu, X. (2020). 2lpiRNApred: A two-layered integrated algorithm for identifying piRNAs and their functions based on LFE-GM feature selection. *RNA Biology*, 17(6), 892–902. <https://doi.org/10.1080/15476286.2020.1734382>.
22. Wu, J., Liu, X., Han, L., Nie, H., Tang, Y., Tang, Y., Song, G., Zheng, L., & Qin, W. (2022). Small RNA sequencing revealed aberrant piRNA expression profiles in deciduas of recurrent spontaneous abortion patients. *BIOCELL*, 46(4), 1013–1023. <https://doi.org/10.32604/biocell.2022.016744>.
23. Khan, S., Khan, M., Iqbal, N., Amiruddin Abd Rahman, M., & Khalis Abdul Karim, M. (2022). Deep-piRNA: Bi-Layered Prediction Model for PIWI-Interacting RNA Using Discriminative Features.

Computers, Materials & Continua, 72(2), 2243–2258.
<https://doi.org/10.32604/cmc.2022.022901>.

24. Cai, A., Hu, Y., Zhou, Z., Qi, Q., Wu, Y., Dong, P., Chen, L., & Wang, F. (2022). PIWI-Interacting RNAs (piRNAs): Promising Applications as Emerging Biomarkers for Digestive System Cancer. *Frontiers in Molecular Biosciences*, 9, 848105. <https://doi.org/10.3389/fmolb.2022.848105>.



KARACHI INSTITUTE OF ECONOMICS AND TECHNOLOGY



PUBLISHED BY

College of Computing and Information
Sciences, KIET
cocis.kiet.edu.pk
kjcis.kiet.edu.pk
kjcis@kiet.edu.pk

Main Campus

PAF Airmen Academy
Korangi Creek
Tel: 35091114 - 7
Cell: 0336-2508284 - 85

City Campus

Shahra-e-Faisal Site
28-D, Block 6, P.E.C.H.S.
Tel: 34546872, 34532182
Cell: 0336-2508286 - 87

City Campus

North Nazimabad Site
F-103, Block F, Nazimabad
Tel: 36628381, 36679314
Cell: 0336-2444191-92

www.kiet.edu.pk